Advant® OCS
with Master Software

PROFIBUS-DP
Advant® Controller 400 Series

User’s Guide

3BSE 016 040R101 Rev A

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions.

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant is a registered trademark of ABB Asea Brown Boveri Ltd., Switzerland.

Advant Fieldbus is a trademark of ABB Asea Brown Boveri Ltd., Switzerland.

LONWORKS is a registered trademark of Echelon Corporation, U.S.

PROFIBUS is a trademark of PROFIBUS International (P.I.).

Westermo is a registered trademark of Westermo Teleindustri AB, Sweden.

Siemens and all Siemens-products mentioned in this publication are trademarks of Siemens AG.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products AB. ABB Automation Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products AB’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

CE MARKING

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 73/23/EEC.

Copyright © ABB Automation Products AB 2000.

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 Before You Begin 1-1
1.2 Equipment Requirements 1-1
1.3 How to Use This Book 1-1
1.4 Conventions 1-2
1.5 Related Documentation 1-2
1.6 Terminology 1-3
1.7 Product Overview 1-4

Chapter 2 - Installation

2.1 Site Planning 2-1
2.2 Wiring 2-2
  2.2.1 Bus Cable 2-3
  2.2.2 Shielding and Grounding 2-4
2.3 Bus Connectors and Pin Assignments 2-5
  2.3.1 Bus Termination 2-6
  2.3.2 Shielding and Grounding 2-7
2.4 Connecting CI541V1 2-8
  2.4.1 LED Indication 2-10

Chapter 3 - Configuration

3.1 General 3-2
3.2 PROFIBUS I/O Dimensioning 3-3
3.3 Creation and Configuration of CI541 Data Base Element 3-4
  3.3.1 Configuration of CI541 Data Base Element 3-5
    3.3.1.1 Configuration of CI541 Base Part 3-5
    3.3.1.2 Configuration of CI541 Bus Parameters 3-6
3.4 Creation and Configuration of PBSD Data Base Element 3-7
  3.4.1 Configuration of PBSD Data Base Element 3-8
    3.4.1.1 Configuration of PBSD Slave Parameters 3-8
    3.4.1.2 Configuration of PBSD User Parameters 3-10
    3.4.1.3 Configuration of PBSD Configuration Data 3-12
3.5 Creation and Configuration of PBS Data Base Element 3-15
  3.5.1 Configuration of PBS Data Base Element 3-16
    3.5.1.1 Adding a new PBS Data Base Element 3-16
    3.5.1.2 Configuration of PBS Data Base 3-17

CONTENTS (continued)

Chapter 4 - Application Building

4.1 General 4-1
4.2 Understanding VALUENO and Identifier Formats 4-2
  4.2.1 Understanding the VALUENO Terminal 4-2
    4.2.1.1 Diagnostic Data 4-3
  4.2.2 Understanding Identifier Formats 4-4
    4.2.2.1 Understanding Simple Identifier Format 4-5
    4.2.2.2 Understanding Special Identifier Format 4-9
    4.2.2.3 Understanding Special Identifier Format DP/V1 4-10
4.3 Receive Data 4-15
  4.3.1 Configuration of PB-R PC Element 4-16
4.4 Send Data 4-17
  4.4.1 Configuration of PB-S PC Element 4-18
4.5 Receive Diagnostic Data 4-20
  4.5.1 Configuration of PB-DIAG PC Element 4-21

Chapter 5 - Fault Tracing

5.1 To do First 5-1
5.2 Hardware Indicators 5-2
5.3 Data Base Elements Indicators 5-2
  5.3.1 CI541 Data Base Element Indicators 5-3
  5.3.2 PROFIBUS Slave Data Base Element Indicators 5-5
5.4 PC Element Indicators 5-7
5.5 PROFIBUS related System Messages 5-7

Appendix A - Technical Data

A.1 Hardware Module A-1
A.2 Performance A-3

Appendix B - PROFIBUS-DP

B.1 PROFIBUS-DP Technical Overview B-1

Chapter 1 Introduction

1.1 Before You Begin

This manual describes the usage of PROFIBUS interface as implemented in Advant Controller 400 Series.

1.2 Equipment Requirements

To use the PROFIBUS interface in Advant Controller 400 Series, you must have the following:

- Submodule carrier SC510 or SC520
- CI541V1 communication interface

1.3 How to Use This Book

This book is organized into chapters and sections as shown in Figure 1-1.

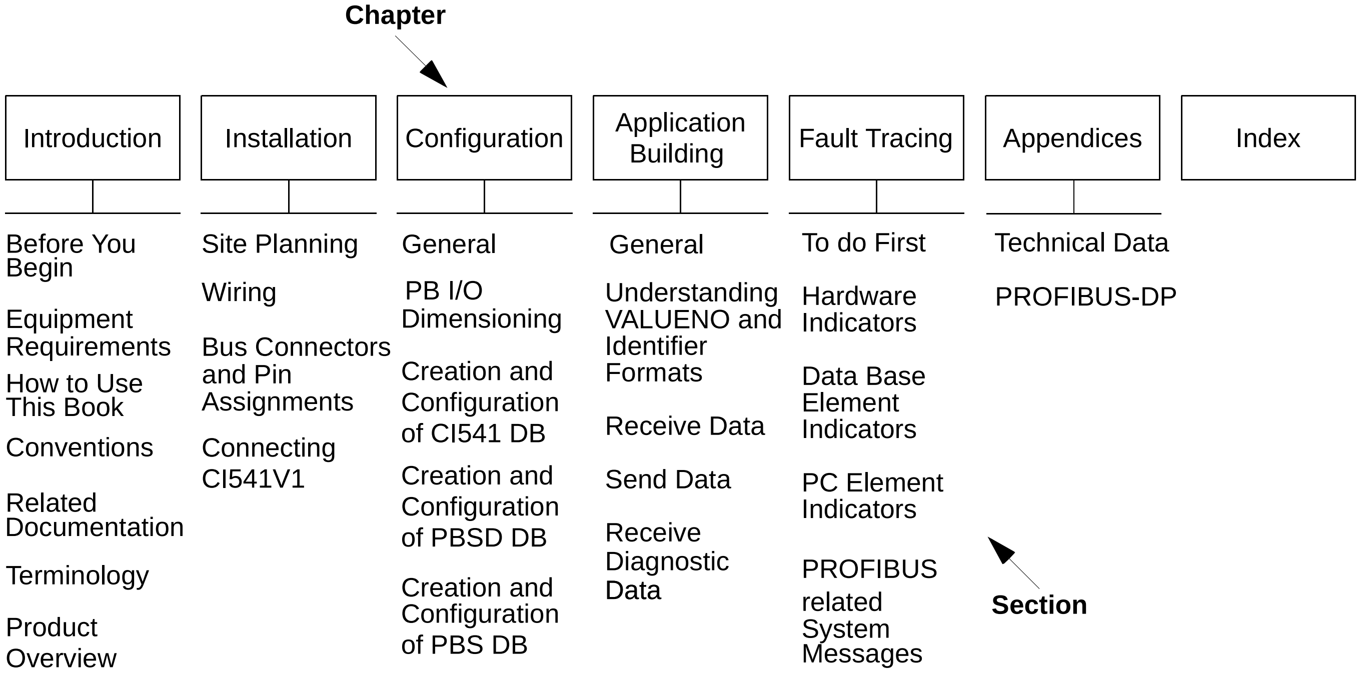


Figure 1-1. Organization of Manual

1.4 Conventions

The following conventions are used throughout this manual for the presentation of material:

- Terminal names are formatted as **bold**, except in figures where they are plain courier.
- GSD keywords are formatted as *italic*.

1.5 Related Documentation

Table 1-1 lists all documentation related to PROFIBUS-DP

Table 1-1. Related Documentation

Title	Description
Advant Controller 410 User's Guide	This manual provides information about the Advant Controller 410 system, its capabilities and its limitations.
Advant Controller 450 User's Guide	This manual provides information about the Advant Controller 450 system, its capabilities and its limitations.
Advant OCS Installation Rules User's Guide	User's Guide for compliance with the EMC requirements.
AMPL Configuration Advant Controller 400 Series Reference Manual	Instructions for the configuration and application programming of Advant Controller 410/450 systems using Advant Station 100 Series ES. Commands for diskette handling are described in the User's Manual on the tool concerned.
Data Base Elements AC 400 Series Reference Manual	Data for all data base elements in Advant Controller 410/450.
PC Elements AC 400 Series Reference Manual	Data for all PC elements in Advant Controller 410/450.
PROFIBUS Library Editor User's Guide	Describes the PROFIBUS-DP configuration tool for AC 400 Series and AC 100 Series.
PROFIBUS Technical Guideline Installation Guideline for PROFIBUS-DP/FMS.	Describes common procedures to follow when specifying a PROFIBUS installation.
www.profibus.com	PROFIBUS Internet site.

1.6 Terminology

The following is a list of terms associated with PROFIBUS that you should be familiar with.

Term	**Description**
Advant OCS	Short for Advant Open Control System, with Master software, throughout this manual.
Bus	A bus is the physical connection between all nodes, without any protocol.
Bus Master	The communication interface that currently controls the bus traffic.
CI541	Data base element.
CI541V1	PROFIBUS Communication Interface.
Communication Interface	A communication interface is a device that can communicate on the PROFIBUS communication link.
DB	Data Base.
EN 50170	Containing european standard for PROFIBUS, World FIP, and P-Net. Successor to German national standard DIN 19245.
GSD file	Description file. A text file containing a description of the PROFIBUS device, with a predetermined syntax. It is delivered with the device or can be obtained from the manufacturer.
PBS	PROFIBUS Slave.
PBSD	PROFIBUS Slave Description.
PB-DIAG	PROFIBUS-Diagnostic PC Element.
PB-R	PROFIBUS-Read PC Element.
PB-S	PROFIBUS-Send PC Element.
PROFIBUS	PROcess FIeld BUS. When mentioned in this manual it always refers to PROFIBUS-DP.

1.7 Product Overview

PROFIBUS specifies the technical and functional characteristics of a serial fieldbus system with which decentralized I/O devices can be networked together from the field level to the cell level. PROFIBUS distinguishes between master devices and slave devices.

Master devices determine the data communication on the bus. A master can send messages without an external request when it holds the bus access rights (the token). Masters are also called active stations in the PROFIBUS protocol.

Slave devices are peripherial devices. Typical slave devices include input/output devices, valves, drives and measuring transmitters. They do not have bus access rights and they can only acknowledge received messages or send messages to the master when requested to do so. Slaves are also called passive stations. Since they only require a small portion of the bus protocol, their implementation is particularly economical.

This implementation supports PROFIBUS-DP Master (Class 1) according to EN 50170 version 1.0. The implementation also supports a subset of PROFIBUS DP/V1 with following data types:

- Boolean
- Integer8
- Integer16
- Integer32
- Unsigned8
- Unsigned16
- Unsigned32
- Floating Point
- Visible String
- Octet String

In addition this implementation also supports PROFIBUS-DP Master (Class 2) services:

- Read input, output and diagnostic data from a slave device being configured and controlled by another PROFIBUS-DP master device.

  This additional functionality requires:

  - Advant Controller 450 ∗2.3 (or later) or
    Advant Controller 410 ∗1.5 (or later)
  - CI541V1 communication module with revision "E" or higher, which can be seen by looking at the label attached to the circuit board and finding PR:E.

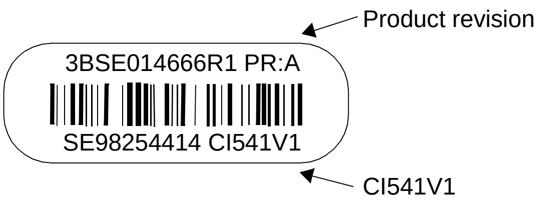


Figure 1-2. Example of Label for Product Revision A (PR:A) of CI541V1

Chapter 2 Installation

This chapter will guide you through the installation of the PROFIBUS communication interface CI541V1 shown in Figure 2-1.

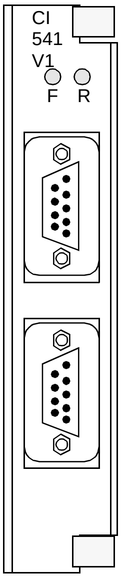


Figure 2-1. CI541V1

The chapter focuses on physical installation of the PROFIBUS, and not on system setup and configuration. Repeaters, connectors, cables and other PROFIBUS products can be found in the product catalog at www.profibus.com.

NOTE

Insertion of the CI541V1 **cannot** be done in a running Advant Controller 410. You first have to power off the Advant Controller 410, insert the CI541V1 and then turn on the power again.

In Advant Controller 450 you first have to unplug the submodule carrier (SC510 or SC520). After that you can insert the CI541V1 into this submodule carrier and finally plug in the submodule carrier again. This means that you do not have to power off the whole Advant Controller 450.

2.1 Site Planning

Please see the *Advant OCS Installation Rules User's Guide* and *Advant Controller 450 User's Guide* or *Advant Controller 410 User's Guide* for guidelines on grounding, power supply and so on.

A PROFIBUS-DP bus installation consists of linked sections forming a chain, that is, a linear bus topology.

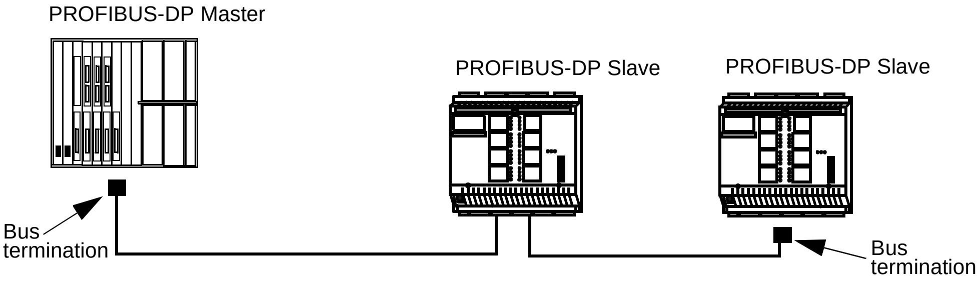


Figure 2-2. Bus Topology

2.2 Wiring

All devices are connected in a bus structure. You can connect up to 32 stations (masters or slaves) on one segment. If you use more than 32 stations, repeaters (line amplifiers) must be used to connect the individual bus segments.

NOTE

EN 50170: A maximum of 3 (three) repeaters between two PROFIBUS slave or master devices is permissible.

The maximum cable length depends on the transmission speed (see Table 2-1). It should be taken into consideration that the total length includes the sum of the stub lengths. The specified cable length can be increased by the use of repeaters. Examples on repeaters can be found in PROFIBUS product catalog at www.profibus.com.

Table 2-1. Maximum Cable Lengths per Segment

Data transfer rate in Mbit/s	0,0096	0,0192	0,09375	0,1875	0,5	1,5	3	6	12
Segment length in m	1200	1200	1200	1000	400	200	100	100	100
Segment length in feet	3937	3937	3937	3281	1312	656	328	328	328

NOTE

In a PROFIBUS installation, there is only one data transfer rate - the rate specified by the master. In addition, the data transfer rate is limited by the "weakest" or longest bus segment length. According to EN 50170, fieldbus devices don't need to support all available data transfer rates. You should take this into account when choosing the fieldbus devices.

2.2.1 Bus Cable

You can only use one type of media for the PROFIBUS communication interface CI541V1 and that is twisted pair media. Coaxial and optical media are not yet supported. Depending on the application, you can choose freely among different kind of twisted pair media. The only thing you should bare in mind is that the cables comply to the cable parameters in Table 2-2 below.

The bus cable type B specified in EN 50170, is outdated and should no longer be used. You are therefore recommended to only use cable type A. An example of a PROFIBUS cable type A is Siemens LAN cable with order number 6XV1 830 0AH10. The cable type A parameters are specified in Table 2-2.

Table 2-2. Specified Cable Type A Parameters

Parameter	Value
Cable Type	Twisted Pair Cable. 1x2 or 2x2 or 1x4 lines
Core Cross-section	> 0,34 mm^2
Core Diameter	< 0,64 mm
Impedance	135 up to 165 Ohm at a frequency of 3 to 20 MHz
Loop Resistance	< 110 Ohm/km
Operating Capacity	< 30 pF/m
Shielding	CU shielding braid or shielding braid and shielding foil
Signal Attenuation	max. 9 dB over total length of line section

The two data lines in Figure 2-8 are designated A and B. There are no regulations on which cable core color should be connected to which of the two data terminals on each fieldbus device; the sole requirement is that you ensure that the same core color is connected to the same terminal (A or B) for all stations throughout the entire system (across all bus segments). If the data transfer cable has data wires with red and green insulation, then the following assignment should be used (See also Figure 2-6):

- Data cable core A - green
- Data cable core B - red

2.2.2 Shielding and Grounding

NOTE

It is very important that the shielding of the bus cable has contact with the metal inside the bus connector (see Figure 2-3), to ensure safe grounding.

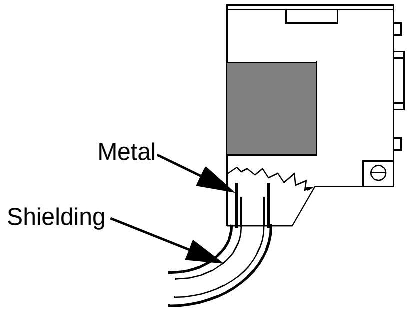


Figure 2-3. Bus Connector and Shielding

Different ground voltages within the same plant are often due to the different main power supplies for the machines within the plant. The grounds of the fieldbus devices are interconnected with the cable shield of the bus cable. With fieldbus devices using grounded interfaces, the ground potential difference can still cause problems if they are grounded to different parts of the plant even though the cable shield is connected at both ends of the cable. This can result in high potential equalization currents for which the electronic components are not designed. The ground potential differences can be reduced by laying a separate potential equalization cable.

NOTE

It is important to have potential equalization if you have a direct connection between two systems as seen in Figure 2-4.

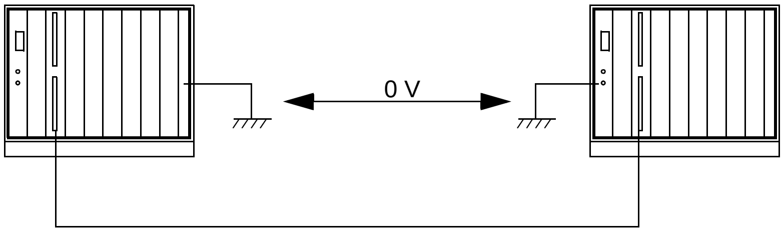


Figure 2-4. Direct Connection between two Systems

If it is necessary to interconnect two relatively long bus segments and significant potential differences between the bus segments are likely, it is recommended to use a non-grounded connection using a repeater/modem to provide galvanic isolation between the two bus segments.

Also in those cases when you can not guarantee potential equalization it is recommended to connect modems as shown in Figure 2-5.

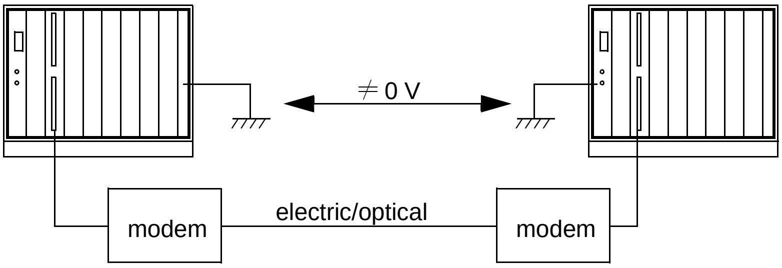


Figure 2-5. Direct Connection with Modems

Recommended modems are Westermo RA-48 or Westermo MD-63. Others can be found at www.profibus.com.

2.3 Bus Connectors and Pin Assignments

Use a bus connector to interconnect the bus cable and the station. The design of the PROFIBUS connector should ensure that stations can be connected to and unplugged from the PROFIBUS network while the bus is operating and without interfering with other stations. This is done by passing the data signals through the connector allowing the data signals to pass to/from the device and on to the next device.

EN 50170 recommends that you use a 9-pin SUB-D connector.

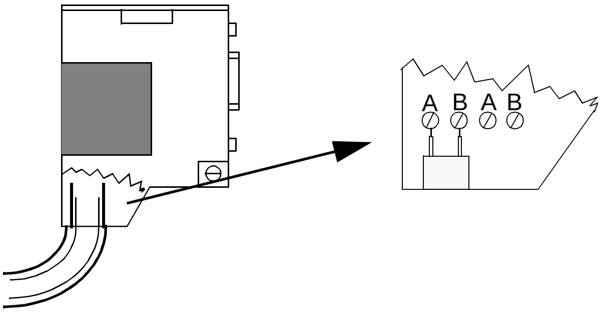


Figure 2-6. Connecting the Bus Cable to the Bus Connector

In Figure 2-7 you can see the pin assignment of the bus connector.

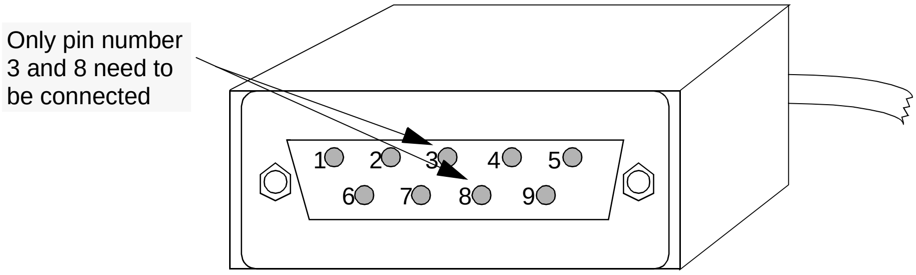


Figure 2-7. 9-pin Male SUB-D Connector

2.3.1 Bus Termination

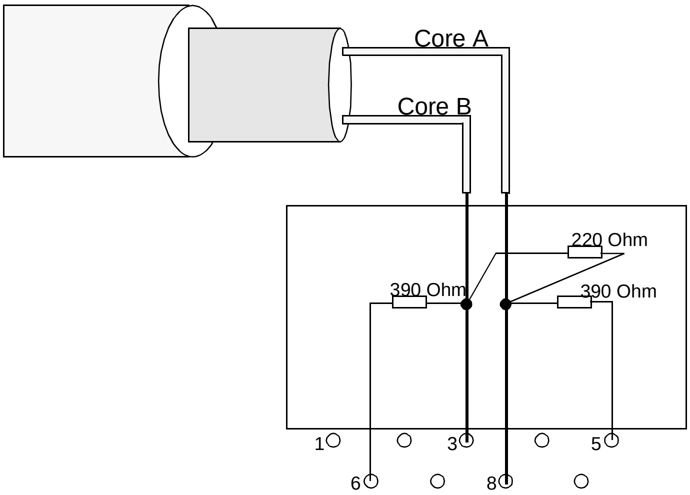


Figure 2-8. Cable Termination of Cable A to EN 50170

In order to minimize cable reflections and ensure a defined noise level on the data transfer lines, you must terminate at both far ends of the bus with the terminating resistor combination shown in Figure 2-8 above. To ensure error-free operation, both bus terminations must always be powered.

NOTE

You should be aware of that termination can be built into the connector.

2.3.2 Shielding and Grounding

If a repeater or fieldbus device is installed within a control cabinet, the cable shield of the incoming bus cable should be electrically connected to a grounding rail as close as possible to the cable lead-through using a shield grounding clamp or similar (see Figure 2-9). The cable shield should continue within the cabinet to the fieldbus device and be connected there in accordance with the manufacturer's instructions. The following installation guidelines should be observed. Ensure that the case of the device and also the control cabinet in which the fieldbus device is mounted have the same ground potential by providing a large-area metallic contact to ground (use for example galvanized steel to ensure a good connection). Grounding rails should not be attached to painted surfaces. If the above measures are observed, electromagnetic interference is diverted through the cable's shield.

Refer to Section 2.4, Connecting CI541V1.

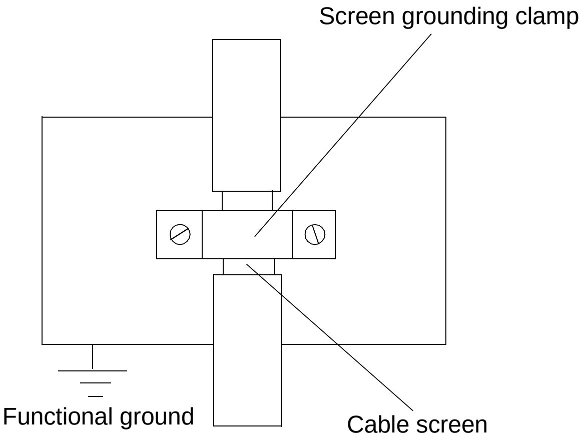


Figure 2-9. Shield Connection of PROFIBUS Cables to a Conductive Surface

2.4 Connecting CI541V1

Now that you have gained information about cable types, pin assignments and termination you are ready to connect the CI541V1 to the network. First is a connection (Figure 2-10) assuming CI541V1 at the far end on the bus, where termination also **must** be done.

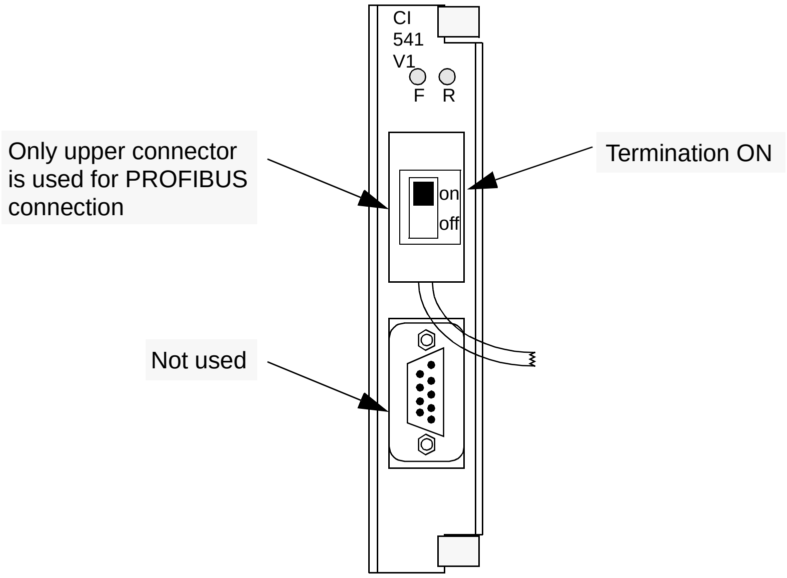


Figure 2-10. Connecting CI541V1 at far End of Bus

NOTE

For the CI541V1 which is always mounted in a cabinet, the recommended solution for shielding is described in *Advant Controller 450* and *Advant Controller 410 User's Guides* under heading Grounding of Communication Cable Shields in Chapter Installation, see Figure 2-11. Pin 1 cannot be used for shielding. For site planning it is recommended to read the *PROFIBUS Technical Guideline Installation Guideline for PROFIBUS-DP/FMS.*

From the upper connector you should connect the communication cable to a capacitive decoupling unit which is mounted on the horizontal mounting bar. See Figure 2-11. See also *Advant Controller 450 User's Guide* or *Advant Controller 410 User's Guide.*

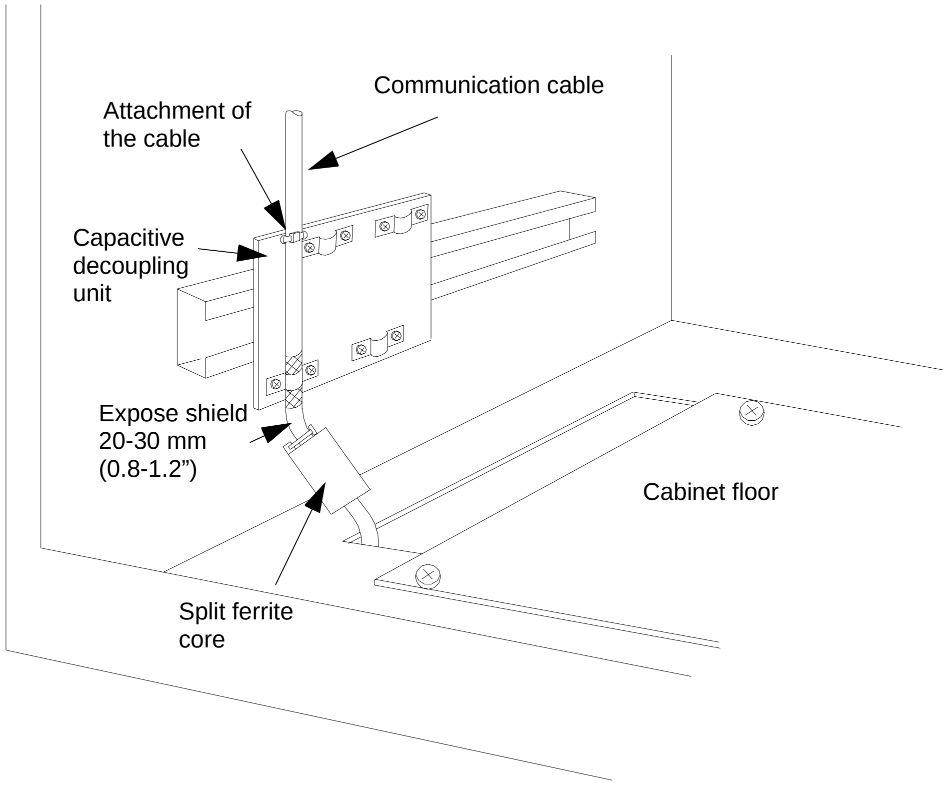


Figure 2-11. Capacitive Decoupling Unit

Next figure (Figure 2-12) is a connection assuming CI541V1 mounted somewhere in the middle of the bus, where **no** termination should be done.

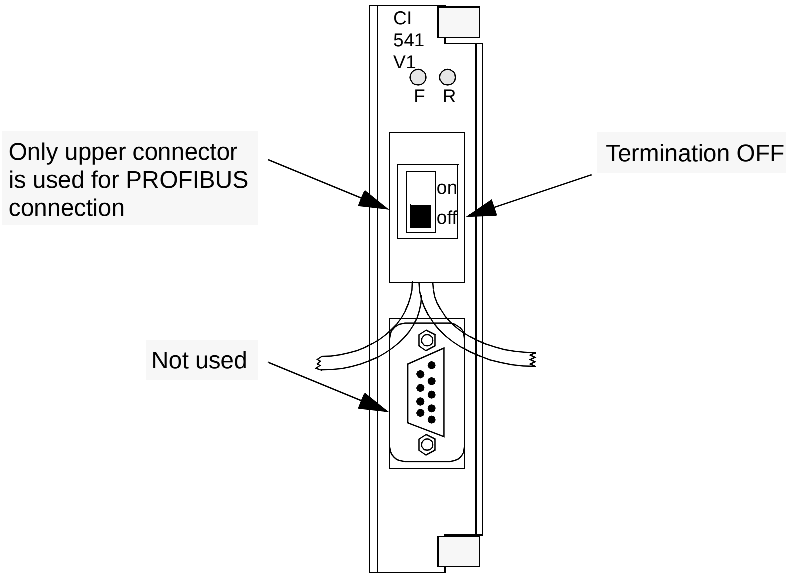


Figure 2-12. Connecting CI541V1 in middle on the Bus

2.4.1 LED Indication

When you are done with installation check LED's and see what color they show:

- Red: selftest. If the bus has been configured and the card is OK it will turn to green after a few moments.
- If the red light continues to shine then something is physically wrong with the card. See Chapter 5, Fault Tracing
- Nothing: the system is starting up, or the bus is not configured. Move on to Chapter 3, Configuration.
- Green: OK. Bus is operating.

Chapter 3 Configuration

This chapter will guide you through the configuration phase. The communication interface can only be operational if the configuration is done correctly.

To set up the data base elements needed to configure the PROFIBUS link, you need to define the following:

- Network structure:
  - Local control network number.
  - Node numbers for all stations on the PROFIBUS communication link.
  - Slaves to be used.
- Characteristics for the line:
  - Transmission speed.
  - PROFIBUS timing.
- PROFIBUS data base elements.

The defining procedure described above is easiest done if you follow the steps below. Each step refers to a section of this chapter.

1. Start your On-line Builder- or Function Chart Builder-tool and add the dimensioning of the PROFIBUS I/O. See Section 3.2, PROFIBUS I/O Dimensioning.
2. Create and configure CI541 data base element. See Section 3.3, Creation and Configuration of CI541 Data Base Element.
3. Create and configure PROFIBUS slave description data base elements for all types of units connected to the PROFIBUS bus. See Section 3.4, Creation and Configuration of PBSD Data Base Element.
4. Create and configure PROFIBUS slave data base elements for all slaves connected to the PROFIBUS bus. See Section 3.5, Creation and Configuration of PBS Data Base Element. At this moment you can either start up your communication link or move on to the next and final step.
5. Build Application PC-program. See Chapter 4, Application Building.

NOTE

The PROFIBUS Library Editor supports this procedure.

For each slave it generates two type circuits for work with the Function Chart Builder. They encapsulate PROFIBUS slave description data base elements that are configured as described in this chapter. The second one encapsulates also the appropriate PROFIBUS slave input, output, and diagnostic PC elements that are described in Chapter 4, Application Building.

The PROFIBUS Library Editor generates also AMPL source code templates for work with the On-line Builder. See *PROFIBUS Library Editor User's Guide*.

3.1 General

The PROFIBUS data base elements are introduced in the Advant Controller 400 series to support distributed I/O on a PROFIBUS bus.

Each PROFIBUS slave is represented by two data base elements in the controller. The PROFIBUS slave data base element contains general slave parameters and the PROFIBUS slave description data base element contains PROFIBUS slave description. The PROFIBUS slave description data base element normally contains data that can be taken directly from the GSD file.

The CI541 data base element and the PROFIBUS slave data base elements are stored in existing logical files in the data base and the PROFIBUS slave descriptions are stored in separate logical files that need to be dimensioned in the data base.

Data base element types, item designations, and call names are according to Table 3-1.

Table 3-1. PROFIBUS Data Base Element Types

Data Base Element Type	Item Designation	Call Name
CI541V1 Communication Interface	PBs	CI541
PROFIBUS Slave	PBSx	PBS
PBS Description	PBSDx	PBSD

In the Advant Controller 400 Series there has to be a way to uniquely address an I/O station or I/O module. This means that the combination **BUS** and **STATION** has to be unique for fieldbuses of the type PROFIBUS and Advant Fieldbus 100.

The **BUS** terminal on the PROFIBUS slave data base element specifies on what communication link the object is located.

The **STATION** terminal on the PROFIBUS slave data base element specifies the PROFIBUS slave address. This is normally set by rotary switches on the slave itself.

NOTE

Address terminals can not be modified when the data base element is implemented (**IMPL** = 1) and the system is in operational mode.

NOTE

Because Advant Fieldbus 100 buses, LONWORKS, and PROFIBUS buses are sharing resources in the system there is a strong dependency between them. The total number of fieldbuses (Advant Fieldbus 100, LONWORKS, and PROFIBUS) is set to be maximum 8 in Advant Controller 450 and maximum 4 in Advant Controller 410.

3.2 PROFIBUS I/O Dimensioning

The first thing you should do in the configuring phase is to add the following dimensioning data. Remember that it is better to overestimate the number of devices. This because the number of dimensioned devices can not be increased later in operation mode.

We recommend you to follow the step-by step guiding below but it is not mandatory.

1. Enter number of PROFIBUS FIELDBUSES.
2. Enter number of PROFIBUS SLAVES.
3. Enter number of PROFIBUS SLAVE DESCRIPTIONS.

NOTE

Dimensioning of the number of PROFIBUS SLAVE DESCRIPTIONS, in step number 3 above, is depending on the amount of data (**UPRMDx** and **CFGDx** in data base element PBSD). Have you planned to use of a large amount of data, it is sometimes necessary to overestimate the number of PROFIBUS SLAVE DESCRIPTIONS.
The sum of the terminals UPRMDLEN and CFGDLEN cannot exceed 217 data bytes.

Add the desired numbers and move on to next section, Section 3.3, Creation and Configuration of CI541 Data Base Element.

3.3 Creation and Configuration of CI541 Data Base Element

This section is a description on how to create and configure a CI541 data base element.

Understanding CI541 Data Base Element

The CI541 data base element specifies a communication interface that supports PROFIBUS. The data base element contains configuration as well as diagnostic information of the physical CI541V1 submodule.

Create data base element for CI541. You should get a view like Figure 3-1 on the screen.

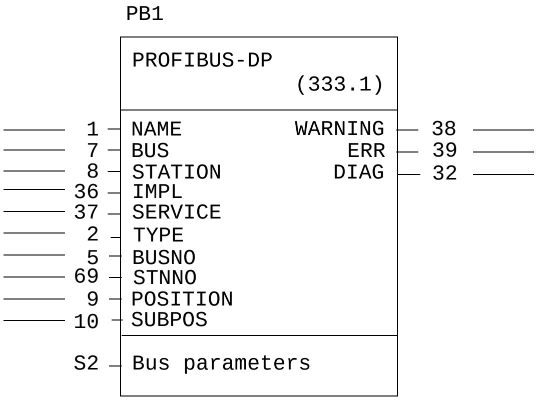


Figure 3-1. CI541 Data Base Element Base Part

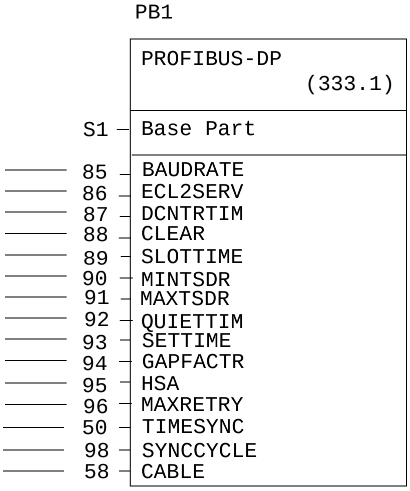


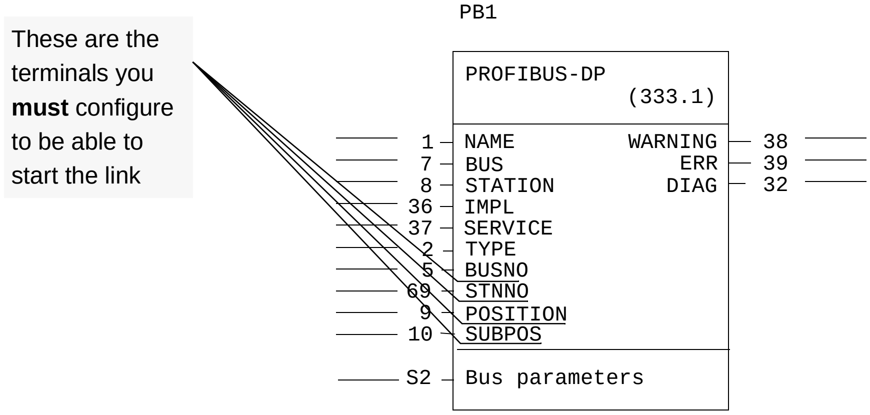
Figure 3-2. CI541 Data Base Element Bus Parameters

In the two figures above (Figure 3-1 and Figure 3-2) you can see an overview of the data base element CI541. In next section (Section 3.3.1, Configuration of CI541 Data Base Element) we will guide you through the configuration of CI541 data base element.

3.3.1 Configuration of CI541 Data Base Element

This section is intended for you who want to start up the communication link quickly with the given default values or feel a little bit unsure of how to configure a PROFIBUS device. In Section 3.3.1.1, Configuration of CI541 Base Part we take a look at base part and in Section 3.3.1.2, Configuration of CI541 Bus Parameters we take a look at bus parameters. If you want more information or want to change the given default values, we recommend you to (also) read *Data Base Elements Advant Controller 400 Series Reference Manual* which also contains a complete list over the terminals discussed in this section.

3.3.1.1 Configuration of CI541 Base Part



These are the terminals you **must** configure to be able to start the link

Figure 3-3. CI541 Data Base Element

In this section you will add values to the terminals that must be configured to be able to start up the communication link. We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **BUSNO**. The bus number, 1...255. The Advant Fieldbus 100 bus and the PROFIBUS bus must have different bus numbers within one Advant Controller. This terminal can not be changed when **IMPL** = 1.
2. Enter **STNNO**. The station number representing the controller on the fieldbus, 0...125. Any number can be chosen. Bare in mind that slaves are also considered stations and therefore it is important that every station has a unique station number. This terminal can not be changed when **IMPL** = 1.
3. Enter **POSITION**. The position of the submodule carrier housing the CI541V1 submodule.
4. Enter **SUBPOS**. The subposition number of the submodule carrier housing the CI541V1 submodule.

Example of a configured CI541 data base element (base part):

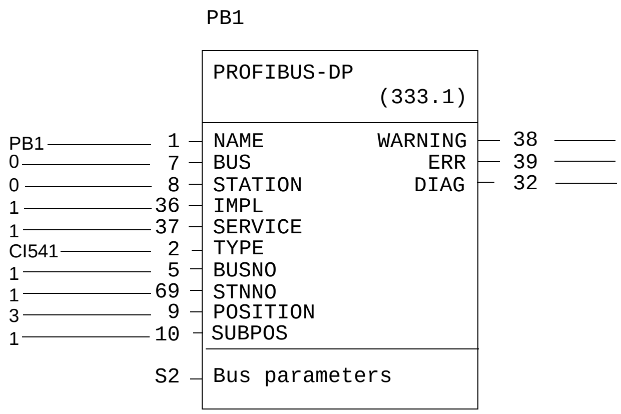


Figure 3-4. Example of a configured CI541 Data Base Element

5. You have finished configuring CI541 base part. Move on to next section, Section 3.3.1.2, Configuration of CI541 Bus Parameters.

3.3.1.2 Configuration of CI541 Bus Parameters

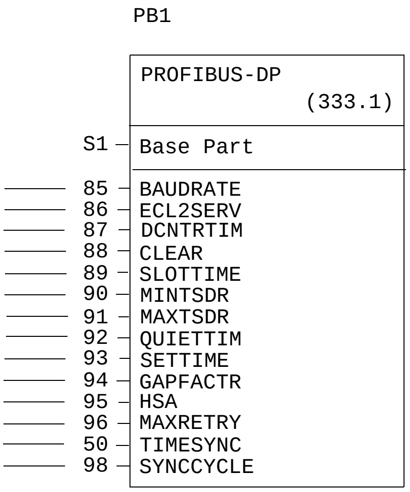


Figure 3-5. CI541 Data Base Element Bus Parameters

We strongly recommend that you accept the given default values. However, these depend on the actual PROFIBUS devices in your application. Please consult the manufacturer's documentation to determine whether the standard values are acceptable. If you want to change the given default values see *Data Base Elements Advant Controller 400 Series Reference Manual*.

3.4 Creation and Configuration of PBSD Data Base Element

This section is a description on how to create and configure a PROFIBUS slave description data base element.

Understanding the PROFIBUS Slave Description Data Base Element:

The PROFIBUS slave description data base element contains descriptions of the PROFIBUS Slave. Normally it contains data that can be found in the GSD file. You have to make a PROFIBUS slave description data base element for each type of PROFIBUS Slave you have in the PROFIBUS network.

Create a data base element for PROFIBUS slave description. You should get a view like Figure 3-6 on the screen:

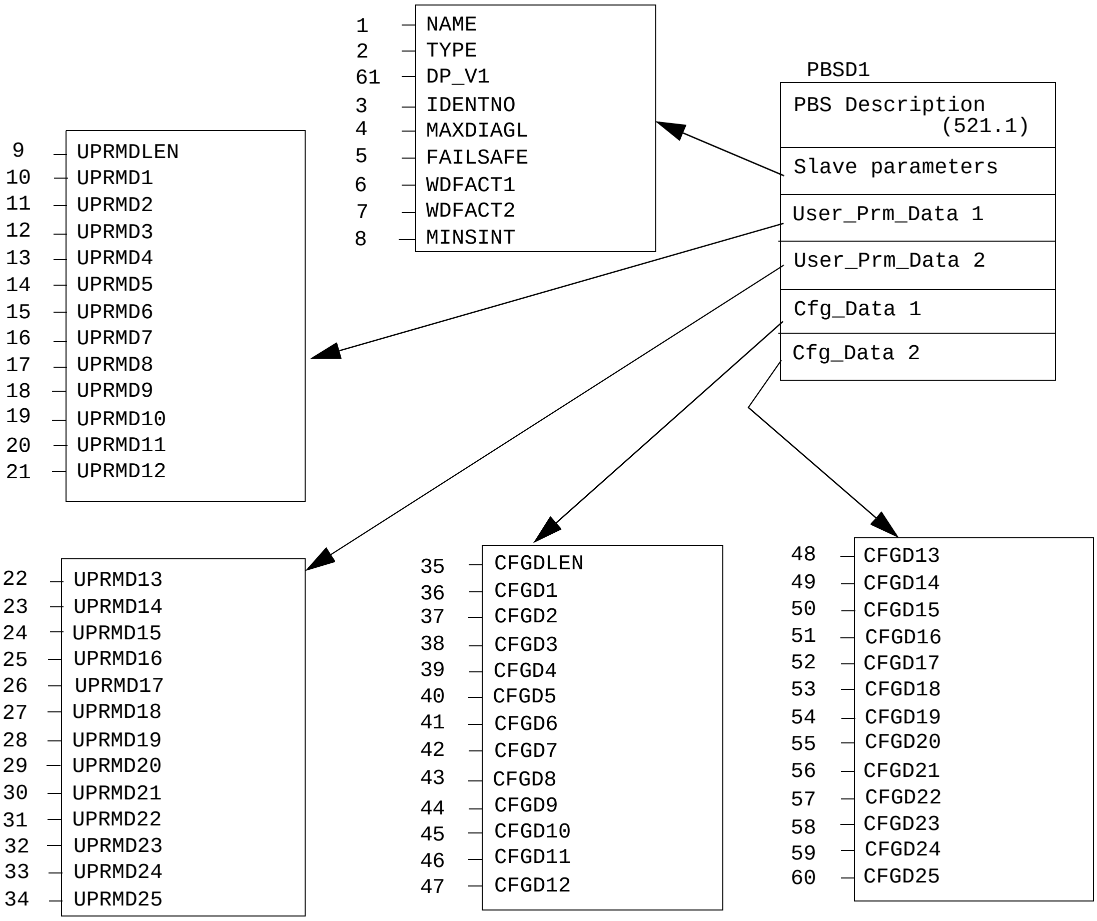


Figure 3-6. PBSD Data Base Element

In next section (Section 3.4.1, Configuration of PBSD Data Base Element) we will guide you through the configuration of this data base element.

3.4.1 Configuration of PBSD Data Base Element

This section is intended for you who want to start up the communication link with the given default values or feel a little bit unsure of how to configure a PROFIBUS slave description data base element. We also discuss some simple examples on how to configure the data base element. In Section 3.4.1.1, Configuration of PBSD Slave Parameters we take a look at slave parameters, in Section 3.4.1.2, Configuration of PBSD User Parameters we take a look at user parameters and in Section 3.4.1.3, Configuration of PBSD Configuration Data we take a look at configuration data. If you want more information or want to change the given default values, we recommend you to (also) read *Data Base Elements Advant Controller 400 Series Reference Manual*, which also contains a complete list of the terminals discussed in this section.

3.4.1.1 Configuration of PBSD Slave Parameters

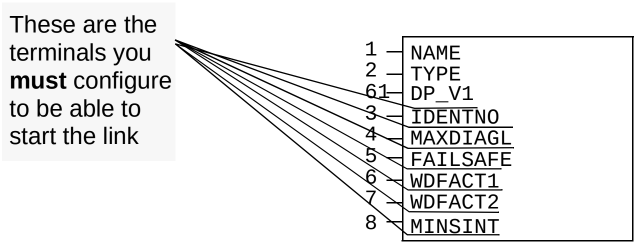


Figure 3-7. Slave Parameters

In this section you will add values to the terminals that must be configured to be able to start up the communication link. We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **DP_V1** = 1, if the slave uses special identifier format **DP_V1** as configuration data (see Section 3.4.1.3, Configuration of PBSD Configuration Data).
2. Enter **IDENTNO** with GSD-keyword: *Ident_Number.*
3. Enter **MAXDIAGL** with GSD-keyword: *Max_Diag_Data_Len*. If you do not find it in the GSD file, try 32.
4. Enter **FAILSAFE** with GSD-keyword: *Fail_Safe*. If you do not find it in the GSD file, try 0.
5. Enter **WDFACT1** and **WDFACT2**. **WDFACT1** x **WDFACT2** x 10 ms = Supervision time out T_{WD}.

Understanding WDFACTx

If a connection between Advant OCS and slave fails, the slave has the options to set all his outputs to 0 after this timer (T_{WD}) has expired (this depends on the type of the slave).

Below in Table 3-2 follows recommended values on **WDFACT1** and **WDFACT2** at different baudrates:

Table 3-2. ***WDFACT1*** *and* ***WDFACT2*** *Recommendations*

Baudrate	WDFACT1	WDFACT2	Supervision Time-out
9.6	10	20	2000
19.2	10	20	2000
93.75	10	20	2000
187.5	10	20	2000
500	10	10	1000
1.5M	5	10	500
3M	2	10	200
6M	1	10	100
12M	1	10	100

6. Enter **MINSINT** with GSD-keyword: *Min_Slave_Interval*.
MINimum Slave INTerval. Minimum time between two poll cycles.
(0...65535 x 0.1 ms. If baudrate 12 Megabaud is used, values >178 are avoided.)

Example of configured PROFIBUS slave description slave parameters:

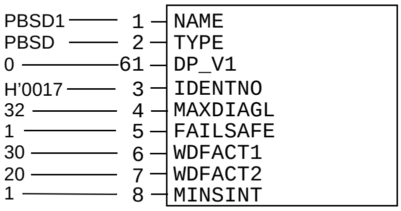


Figure 3-8. Example of configured Slave Parameters

7. You have finished configuring slave parameters. Move on to Section 3.4.1.2, Configuration of PBSD User Parameters.

3.4.1.2 Configuration of PBSD User Parameters

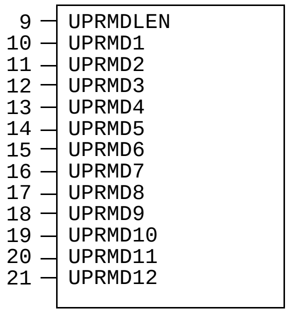


Figure 3-9. User Parameters

The user parameters given in the GSD file are default values and can be changed if you want a different system performance. If doing so we recommend that you consult your PROFIBUS slave manual. We recommend you to follow the step-by-step guiding below, but it is not mandatory.

1. Enter **UPRMDLEN** with GSD-keyword: *User_Prm_Data_Len*. User parameter data length, 0 ... 217. The total length of the following user parameter data.

NOTE

The sum of the terminals **UPRMDLEN** and **CFGDLEN** cannot exceed 217 data bytes.

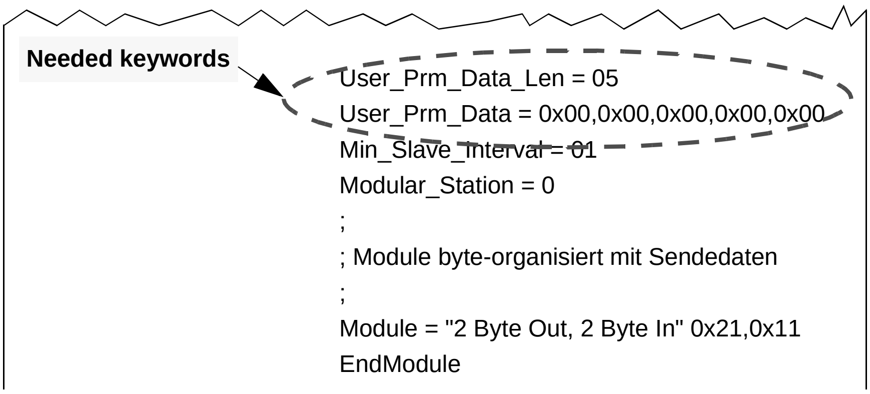
2. Enter **UPRMD1**... **UPRMD25** with GSD-keyword: *User_Prm_Data*. User parameter data. Each terminal contains 10 bytes. See example on next page. Remember that the configuration values are in hexadecimal format.

NOTE

The exact behavior depends on the engineering tool that is used (FCB/ONB), but the ideas are the same.

Example on user parameters:

In this example (Figure 3-10 and Figure 3-11) you can see how you configure the user parameters using a GSD file. The needed keywords from the GSD file are shown in the figure below.



```
User_Prm_Data_Len = 05
User_Prm_Data = 0x00,0x00,0x00,0x00,0x00
Min_Slave_Interval = 01
Modular_Station = 0
;
; Module byte-organisiert mit Sendedaten
;
Module = "2 Byte Out, 2 Byte In" 0x21,0x11
EndModule
```

Figure 3-10. Sample from a GSD File

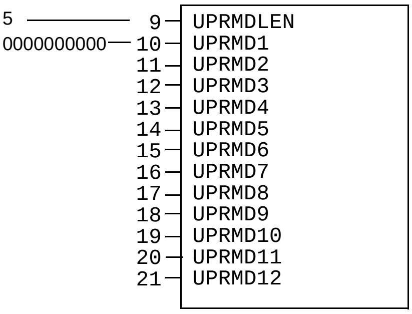


Figure 3-11. Example of configured User Parameters

In Figure 3-11 you can see how to insert values from the GSD file. Observe that you should only insert the numbers directly following after "0x".

NOTE

Depending on the slave, not all **UPRMDx** will be used.

Bare in mind that there is only space for 10 bytes hexadecimal values per terminal (20 characters). If **UPRMDLEN** is > 10 you will have to use several terminals. This is the reason why the user parameters are called **UPRMD1** up to **UPRMD25**.

3. You have finished configuring user parameters. Move on to Section 3.4.1.3, Configuration of PBSD Configuration Data.

3.4.1.3 Configuration of PBSD Configuration Data

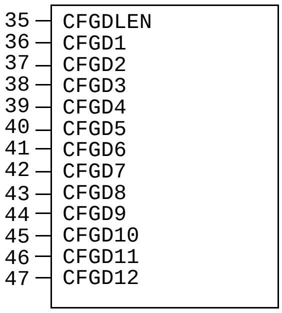


Figure 3-12. Configuration Data

Before reading this section we recommend that you read section Section 4.2, Understanding VALUENO and Identifier Formats in order to get a deeper understanding of configuration data values.

You can, unlike the user parameters, not use other configuration data values than the ones defined in the GSD file. The specified values are needed to configure the device correctly. We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **CFGDLEN** with GSD-keyword: *Module*. Configuration data length, 0 ... 217. The total length of the following configuration data terminals. Unlike **UPRMDLEN** you must calculate this value from the actual configuration. See example on next page.

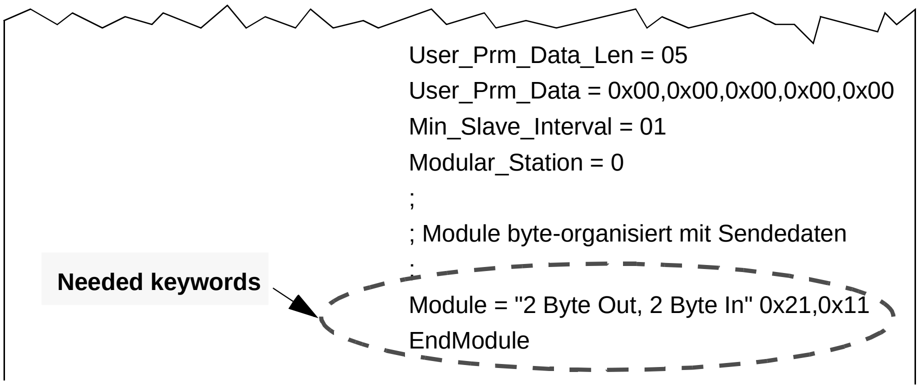
NOTE

The sum of the terminals **UPRMDLEN** and **CFGDLEN** cannot exceed 217 data bytes.

2. Enter **CFGD1**... **CFGD25**. GSD-keyword: *Module*. Configuration data. Each terminal contains 10 bytes. See example on next page. Remember that the configuration values are in hexadecimal format.

Example on configuration data:

In this example (Figure 3-13 and Figure 3-14) you can see an example of how to configure the configuration data using a GSD file. The needed keywords from the GSD file are shown in the figure below..



```
User_Prm_Data_Len = 05
User_Prm_Data = 0x00,0x00,0x00,0x00,0x00
Min_Slave_Interval = 01
Modular_Station = 0
;
; Module byte-organisiert mit Sendedaten
;
Module = "2 Byte Out, 2 Byte In" 0x21,0x11
EndModule
```

Needed keywords

Figure 3-13. Sample from a GSD File

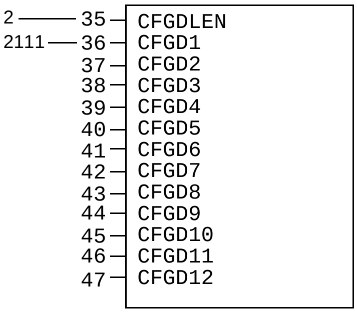


Figure 3-14. Example of configured Configuration Data

In Figure 3-14 you can see how to insert values from the GSD file. Observe that you should only insert the numbers directly following after "0x".

Bare in mind that there is only space for 10 bytes hexadecimal values per terminal (20 characters). If **CFGDLEN** is > 10 you will have to use several terminals. This is the reason why the configuration data are called **CFGD1** up to **CFGD25**.

Configuration Data for Modular Devices

If you have a modular device it is important to remember that if you alter two devices positions in the rack, or add a module, you will also have to make two changes:

- If you have altered positions of two or more devices, you have to change the altered devices configuration data values. See figures below (Figure 3-15, Figure 3-16, Figure 3-17 and Figure 3-18).

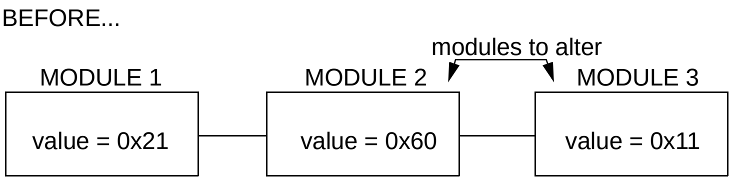


Figure 3-15. Modules before Device Alteration

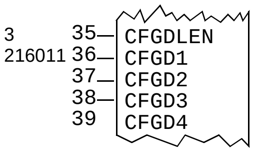


Figure 3-16. Configured Configuration Data before Device Alteration

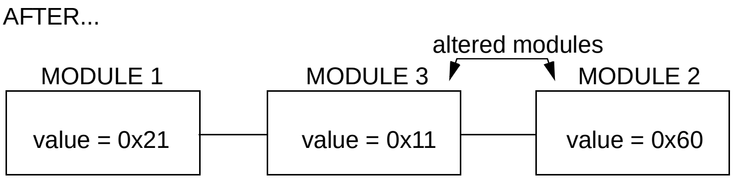


Figure 3-17. Modules after Device Alteration

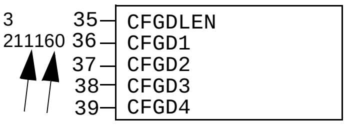


Figure 3-18. Configured Configuration Data after Device Alteration

In the figures above (Figure 3-15, Figure 3-16, Figure 3-17 and Figure 3-18) you can see the changes (marked with arrows) that must be done in the configuration data if you have altered modular devices. The configuration data values are depending on the module they are referring to.

- And if you add or remove a module: recalculate the **CFGDLEN** terminal.

3.5 Creation and Configuration of PBS Data Base Element

This section is a description over how to create and configure a PROFIBUS slave data base element.

Understanding the PROFIBUS Slave Data Base Element

The PROFIBUS slave data base element specifies a PROFIBUS slave on a PROFIBUS bus connected to an Advant Controller 400 series. The data base element contains configuration data and diagnostic information of the PROFIBUS slave.

Create a data base element for PROFIBUS Slave. You should get a view like Figure 3-19 on the screen.

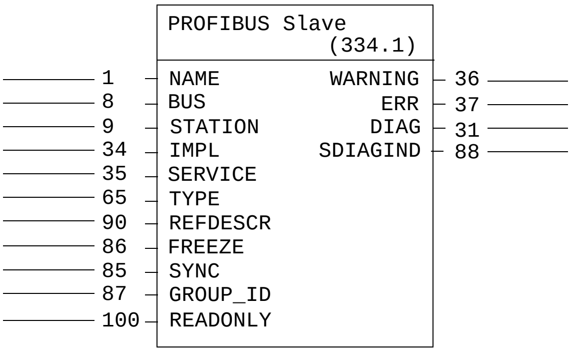


Figure 3-19. PBS Data Base Element

In Figure 3-19 you can see an overview over the PROFIBUS slave data base element. In next section (Section 3.5.1, Configuration of PBS Data Base Element) we will guide you through the configuration of this data base element.

3.5.1 Configuration of PBS Data Base Element

This section is intended for you who want to start up the communication link with the given default values or feel a little bit unsure of how to configure a PROFIBUS Slave data base element. In Section 3.5.1.1, Adding a new PBS Data Base Element we discuss how to add a new PROFIBUS slave data base element and in Section 3.5.1.2, Configuration of PBS Data Base you will be guided on how to configure a PROFIBUS slave data base element. We also discuss some simple examples on how to configure the data base element. If you want more information we recommend you to (also) read *Data Base Elements Advant Controller 400 Series Reference Manual* which also contains a complete list of the terminals discussed in this section.

3.5.1.1 Adding a new PBS Data Base Element

NOTE

This section is only important to read when you are about to add a new PROFIBUS slave data base element to a communication link **after** system start up.

In the PROFIBUS function in the Advant Controller400 series there is a strong dependency between the CI541 **IMPL** terminal and the PROFIBUS slave **IMPL** terminal. New PROFIBUS slaves can only be added when the **IMPL** terminal on the CI541 data base element is set to 0. When adding PROFIBUS slaves, the **IMPL** terminal on the PROFIBUS slave data base element must be set before the **IMPL** terminal on the CI541 data base element, if the PROFIBUS slave shall be activated. If the **IMPL** terminal on the PROFIBUS slave data base element is set to 1 after the **IMPL** terminal on the CI541 data base element, the PROFIBUS slave will remain passive.

The normal procedure to add PROFIBUS slaves is therefore:

1. Set **IMPL** terminal on CI541 data base element to 0.
2. Add the PROFIBUS slave (that is, create and configure data base elements PROFIBUS slave and PROFIBUS slave description).
3. Set the **IMPL** terminal on the PROFIBUS slave data base element to 1 (that is, default value).
4. Set the **IMPL** terminal on the CI541 data base element to 1. This will start the PROFIBUS bus and activate the slaves.

Implementation and activation of PROFIBUS slaves are only performed when CI541 is implemented and activated (**IMPL** = 1).

3.5.1.2 Configuration of PBS Data Base

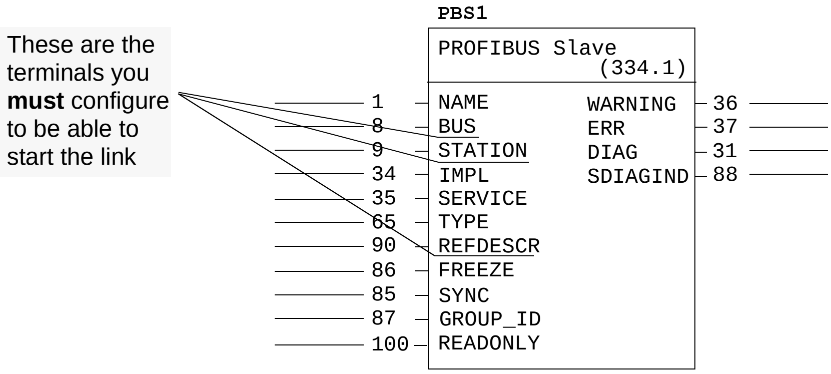

These are the terminals you **must** configure to be able to start the link

Figure 3-20. PBS Data Base Element

In this section you will add values to the terminals that must be configured to be able to start up the communication link.

We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **BUS**. The bus number for this fieldbus, 1...255. This can not be changed when system is in operational mode.
2. Enter **STATION**. The station address representing the slave on the fieldbus, 1...125. This can not be changed when **IMPL** = 1 in operational mode.
3. Enter **REFDESCR**. The reference to the referring PROFIBUS slave description data base element. See example on next page.

NOTE

You must have created a PROFIBUS slave description data base element before you can enter **REFDESCR**.

In Figure 3-21 and Figure 3-22 there is an example on terminal **REFDESCR** where you can see the connection between PROFIBUS slave description data base element and PROFIBUS slave data base element (in the figures marked with a circle around the reference name of the PROFIBUS slave description data base element).

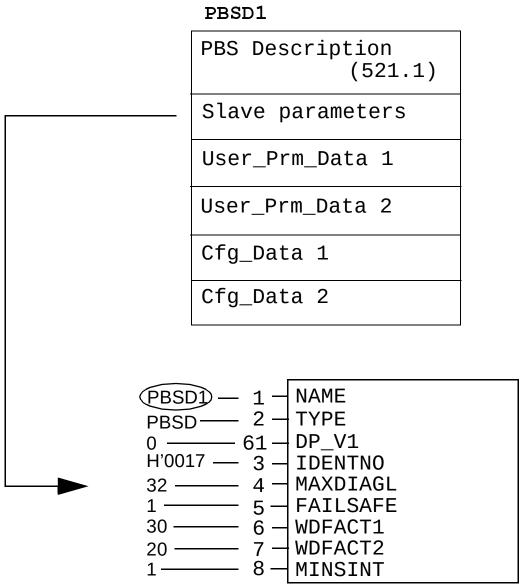


Figure 3-21. Example of a configured PBSD Data Base Element

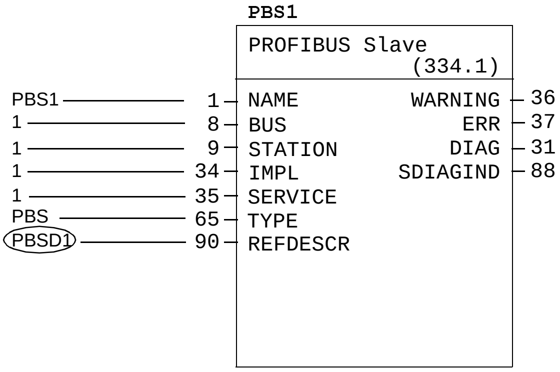


Figure 3-22. Example of a configured PBS Data Base Element

Chapter 4 Application Building

In this chapter we assume that you have knowledge of application programming of Advant Controller 400 Series products. If you are not familiar with this kind of programming, we recommend you to read *AMPL Configuration Advant Controller 400 Series Reference Manual* before you continue to read this chapter.

In this chapter we discuss simple and special identifier formats for configuration data in Section 4.2.2, Understanding Identifier Formats. For more information about configuration data see Chapter 3, Configuration. We will also describe the PC elements in Advant Controller 400 Series used for data exchange with the CI541V1 submodule for PROFIBUS.

NOTE

The PROFIBUS Library Editor supports also PC programming of PROFIBUS slaves.

For each slave it generates two type circuits for work with the Function Chart Builder. They encapsulate PROFIBUS slave description data base elements that are configured as described in Chapter 3, Configuration. They encapsulate also the appropriate PROFIBUS slave input, output, and diagnostic PC elements that are described below.

The PROFIBUS Library Editor generates also AMPL source code templates for work with the On-line Builder. See *PROFIBUS Library Editor User's Guide*.

4.1 General

Process data from a PROFIBUS node are accessed through PC elements. There are three types of PC elements:

- PB-R for input data. See Section 4.3, Receive Data.
- PB-S for output data. See Section 4.4, Send Data.
- PB-DIAG for diagnostic data. See Section 4.5, Receive Diagnostic Data.

In next section, Section 4.2, Understanding VALUENO and Identifier Formats, we discuss the **VALUENO** terminal and identifier formats before we move on to the three PC elements mentioned above. The **VALUENO** terminal is used to index the data number to or from a specific device.

4.2 Understanding VALUENO and Identifier Formats

In this section we discuss the **VALUENO** terminal and the related issue about identifier formats. Section 4.2.2 is included for deeper understanding and not normally needed for the application building and configuration.

4.2.1 Understanding the VALUENO Terminal

In order to understand how to configure the **VALUENO** terminal, belonging on all three types of PC elements, we strongly recommend you to read this section before moving on to the sections where we actually build applications.

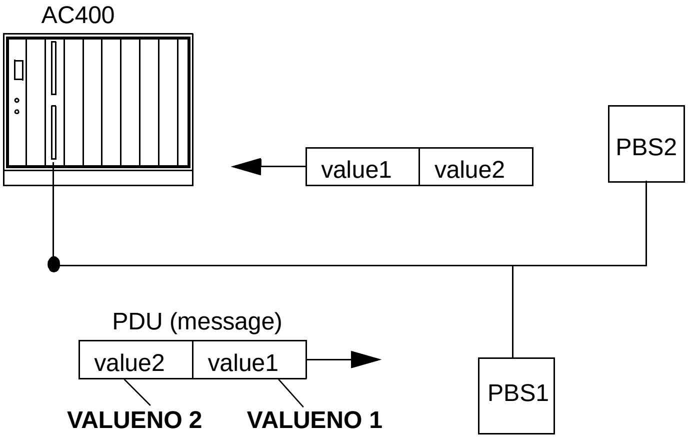


Figure 4-1. Message Sending and Receiving

To explain the **VALUENO** terminal we take a look at Figure 4-1 above. Each PROFIBUS slave sends or receives a number of process values or attributes (of different or equal data types) packed together in a message called a Payload Data Unit (PDU). The PROFIBUS Payload Data Unit for any given slave contains three types of data: input data, output data and diagnostic data. Each of the three parts in Figure 4-2 contains up to 244 [1] values in PROFIBUS-DP data types. In fact, this is the way the PROFIBUS interface provides the data to PM511V1. Internally in the system, process data is stored according to Figure 4-2.

Input data	Output data	Diagnostic data	Status
1 244	1 244	1 244	245

Figure 4-2. PM511V1 View of Process and Diagnostic Data for any given Slave

---

[1] For diagnostic data only slaves with an amount of up to 236 diagnostics bytes are supported.

Please note that in Figure 4-2 there is one additional byte of diagnostic data, the status byte. Process values can be of different types, for example 8 bit (bytes) or 16 bit (words) for PROFIBUS, or even 32 bits (longwords), for example floating point for PROFIBUS-DP/V1. When terminal VALUENO is configured to for example “2”, the referred PROFIBUS device looks for process value number two.

NOTE

If you want to access both “value1” and “value2” in Figure 4-1, you can create two PC elements and access “value1” from PC element number one and “value2” from PC element number two. If “value1” and “value2” are of different data types you have to use two PC elements to access both values. It is possible to access consecutive values with one PC element as long as the values are of the same data type.

For AMPL types I, IL and R, one PC element can be used to access consecutive process data values of the same type.

Data Consistency

PROFIBUS-DP and PROFIBUS-DPVI allow the declaration of consistency data type blocks. The PROFIBUS-DP interface guarantees data consistency only over a singular data type within such a block.

4.2.1.1 Diagnostic Data

The diagnostic data area in Figure 4-2 holds the bytes that the PROFIBUS devices deliver, in order to show their status. The maximum length of this data is 236 bytes. The contents of the 6 (six) first bytes are defined by the PROFIBUS standard. Byte number 7 (seven) and upwards are completely defined by the device’s manufacturer and can be found in the relevant documentation.

The status byte at position 245 in the diagnostics data area delivers information about how the PROFIBUS master views the slave device. Table 4-1 shows the bit definitions in this byte.

Table 4-1. Slave Status as seen by the Advant OCS

Bit	Description
0 ... 3	Device connection status: 0 - Connection up and running. 1 - Error in establishing connection. 2 - Connection aborted. 3 - Device activated. 4 - Device deactivated. 5 - Device not configured.
4	Not used.

Table 4-1. Slave Status as seen by the Advant OCS (Continued)

Bit	Description
5	Device parameter fault indication. (The slave replies with more data than defined in the PBSD MAXDIAGL. The parameter MAXDIAGL must be increased.)
6	Data transfer flag. Process data is transferred between Advant OCS and slave.
7	System diagnostic flag. Slave has reported diagnostics.

4.2.2 Understanding Identifier Formats

This section describes identifier formats and the correspondence to the **VALUENO** terminal and configuration data. Configuration data are described in Section 3.4, Creation and Configuration of PBSD Data Base Element.

Identifier formats are used to understand configuration values found in GSD files, for example:

```
Module = "2 Byte Out, 2 Byte In" 0x21,0x11
EndModule
```

where 0x21 and 0x11 are the configuration values in hexadecimal format.

There are three types of identifier formats:

- Simple identifier format.
- Special identifier format.
- Special identifier format PROFIBUS DP/V1.

The difference between them is the amount of bits and data types they can represent:

  - Simple identifier format can represent 8- and 16 bit data types.
  - Special identifier format can represent 8- and 16 bit data types as well as device specific data formats.
  - Special identifier format DP/V1 can represent 8-, 16- and other data types.

In the following sections, Section 4.2.2.1, Understanding Simple Identifier Format, Section 4.2.2.2, Understanding Special Identifier Format and Section 4.2.2.3, Understanding Special Identifier Format DP/V1 we discuss each one of them.

4.2.2.1 Understanding Simple Identifier Format

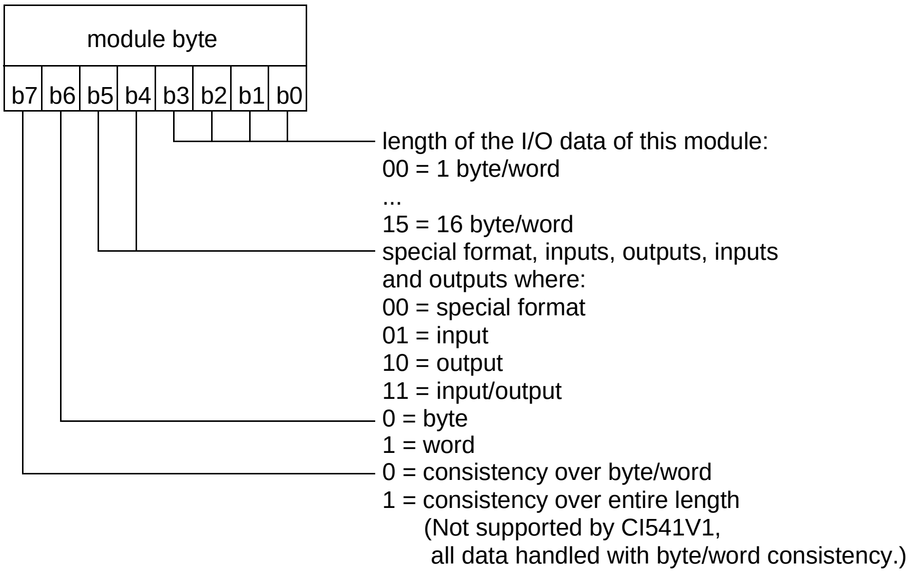

module byte
b7
b6
b5
b4
b3
b2
b1
b0
length of the I/O data of this module:
00 = 1 byte/word
...
15 = 16 byte/word
special format, inputs, outputs, inputs
and outputs where:
00 = special format
01 = input
10 = output
11 = input/output
0 = byte
1 = word
0 = consistency over byte/word
1 = consistency over entire length
(Not supported by CI541V1,
all data handled with byte/word consistency.)

Figure 4-3. Simple Identifier Format

In Figure 4-3 above, you can see a module byte and the meaning of each bit.

NOTE

If bit number 4 and bit number 5 show 0, you have to move on to Section 4.2.2.2, Understanding Special Identifier Format. When showing zeroes it says: "this byte is of a special kind of format, please check special identifier format".

See the following examples to get a deeper understanding of simple identifier format.

Examples on understanding simple identifier format:

Example value 1a: 0x21

If you insert the hexadecimal value 0x21 into the figure above (Figure 4-3), you get the following result (shown in Figure 4-4):

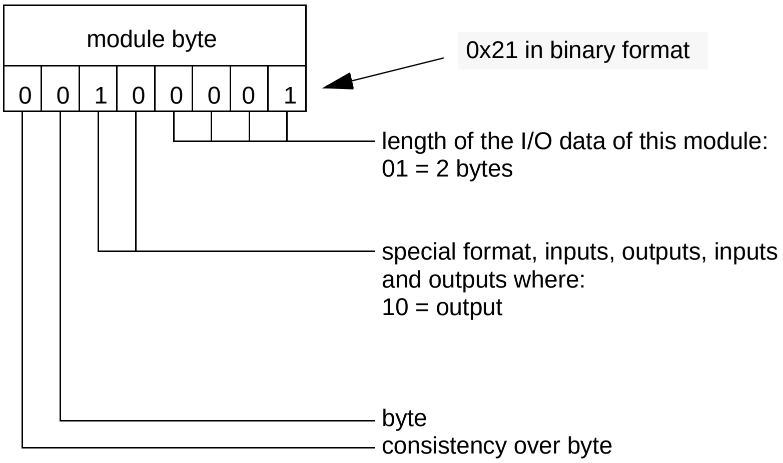


Figure 4-4. Example 1a of Simple Identifier Format

Explanation: The example 1a configuration value for 2 bytes output data.

Example value 1b: 0xE1

If you insert the hexadecimal value 0xE1 into the figure above (Figure 4-3) you get the following result (shown in Figure 4-5):

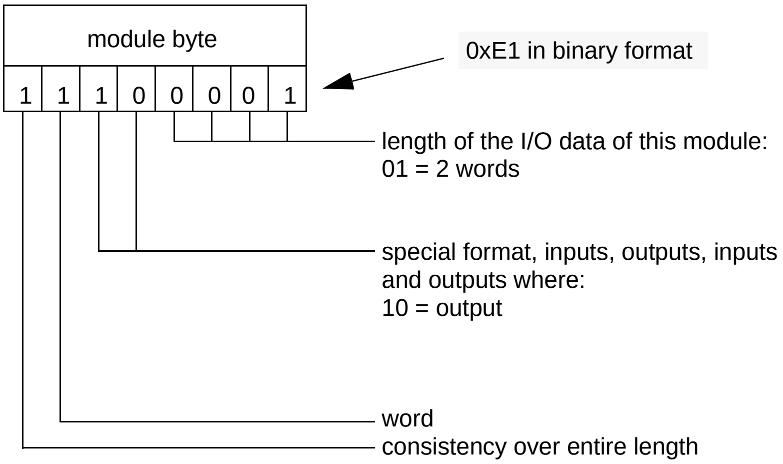

module byte
0xE1 in binary format
1
1
1
0
0
0
0
1
length of the I/O data of this module:
01 = 2 words
special format, inputs, outputs, inputs
and outputs where:
10 = output
word
consistency over entire length

Figure 4-5. Example 1b of Simple Identifier Format

Explanation: The example 1b configuration value for 2 words output data.

See also Section 4.2.1, Understanding the VALUENO Terminal.

Example value 2: 0x11

If you insert the hexadecimal value 0x11 into the figure (Figure 4-3) above, you get the following result (shown in Figure 4-6):

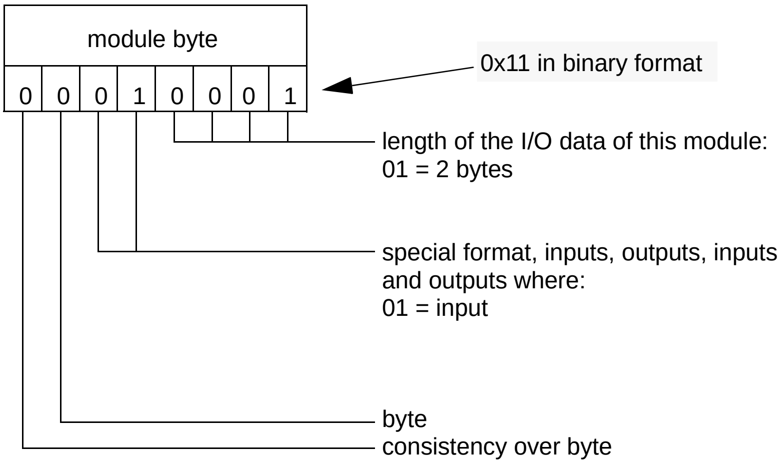


Figure 4-6. Example 2 of Simple Identifier Format

Explanation: The example 2 configuration value for 2 bytes input data.

In the following section, Section 4.2.2.2, Understanding Special Identifier Format, we discuss special identifier format.

4.2.2.2 Understanding Special Identifier Format

In this section we assume that you have read Section 4.2.2.1, Understanding Simple Identifier Format.

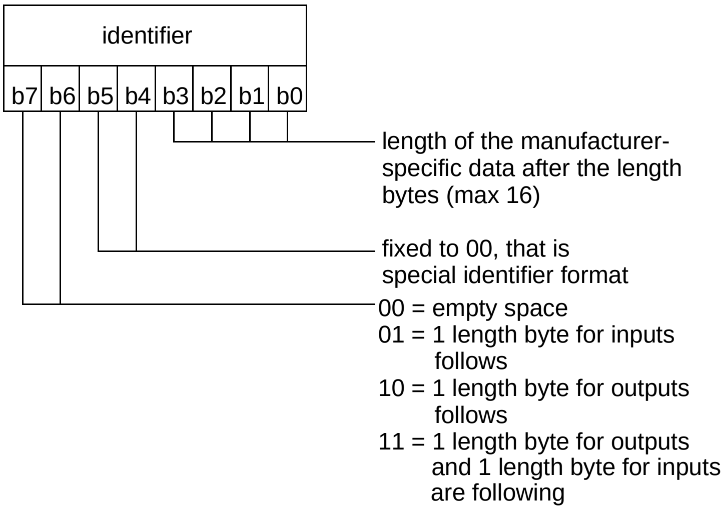

identifier
b7
b6
b5
b4
b3
b2
b1
b0
length of the manufacturer-specific data after the length bytes (max 16)
fixed to 00, that is special identifier format
00 = empty space
01 = 1 length byte for inputs follows
10 = 1 length byte for outputs follows
11 = 1 length byte for outputs and 1 length byte for inputs are following

Figure 4-7. Special Identifier Format, Identifier

In Figure 4-7 above, you can see an identifier byte and the meaning of each bit. The length bytes have the following structure (see Figure 4-8):

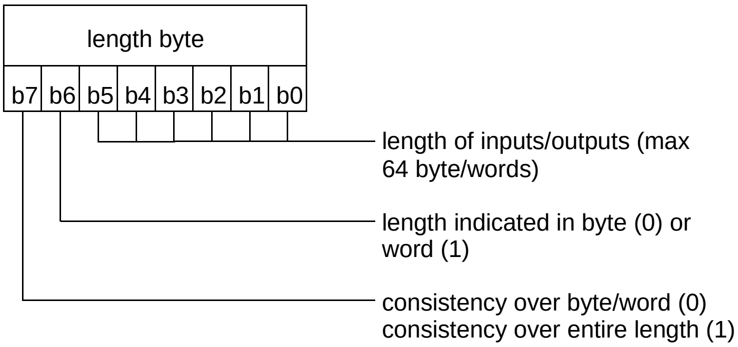

length byte
b7
b6
b5
b4
b3
b2
b1
b0
length of inputs/outputs (max 64 byte/words)
length indicated in byte (0) or word (1)
consistency over byte/word (0)
consistency over entire length (1)

Figure 4-8. Special Identifier Format, Length Byte

The standard special identifier format allows to describe 8- and 16 bit data types. It allows the description of modules having up to 64 bytes/word. For other data types, a third format has to be used (see Section 4.2.2.3, Understanding Special Identifier Format DP/V1).

Examples on special identifier format (Figure 4-9):

									Meaning:
Identifier byte 0	1	1	0	0	0	0	0	1	1 output length byte, 1 input length byte Special Identifier Format, 1 byte user data
Length byte 1	0	1	1	1	1	1	1	1	Consistency over words, 64 words output
Length byte 2	0	1	1	1	1	1	1	1	Consistency over words, 64 words input
User data byte 3	X	X	X	X	X	X	X	X	1 byte user data

Figure 4-9. Example of Special Identifier Format

The example shows the special identifier format for 64 word inputs, 64 word outputs and consistency over words. The "1 byte user data" is device specific, and not described by the standard.

4.2.2.3 Understanding Special Identifier Format DP/V1

In this section we assume that you have read Section 4.2.2.1, Understanding Simple Identifier Format and Section 4.2.2.2, Understanding Special Identifier Format. Please note the correspondence to terminal **DP_V1** on PROFIBUS slave description data base element. See Section 3.4, Creation and Configuration of PBSD Data Base Element.

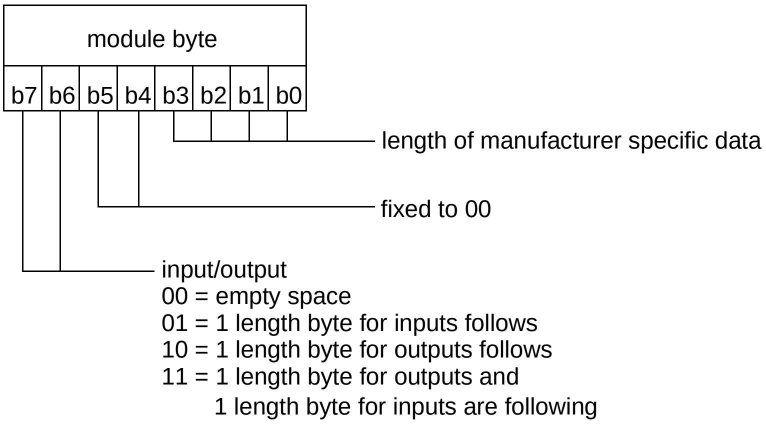

module byte
b7
b6
b5
b4
b3
b2
b1
b0
length of manufacturer specific data
fixed to 00
input/output
00 = empty space
01 = 1 length byte for inputs follows
10 = 1 length byte for outputs follows
11 = 1 length byte for outputs and
1 length byte for inputs are following

Figure 4-10. Special Identifier Format DP/V1

In Figure 4-10 above, you can see a module byte and the meaning of each bit. The length bytes have the following structure (see Figure 4-11):

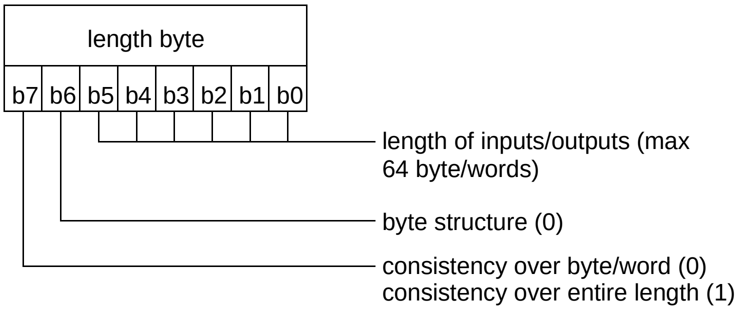

length byte
b7
b6
b5
b4
b3
b2
b1
b0
length of inputs/outputs (max 64 byte/words)
byte structure (0)
consistency over byte/word (0)
consistency over entire length (1)

Figure 4-11. Special Identifier Format DP/V1, Length Byte

The data types are coded into the manufacturer-specific part. Simple data types are coded in a byte which contains the code of the data type. Sequences of data types are coded as a list of simple data types. The sum of the lengths of the data types has to correspond to the input- or output data length. Data types with variable length (visible string, octet string, time of day or time difference) can not be handled in a sequence, but as single element.

The supported data types change with the protocol variant of PROFIBUS:

- Classic PROFIBUS-DP only knows about byte (unsigned8) and word (unsigned16) values. You have to convert these raw formats into data types that suit the physical value, using PC elements.
- PROFIBUS DP/V1 recognizes a large number of different data formats automatically. Table 4-2 and Table 4-3 shows the relation between the various PROFIBUS and Advant OCS data types. Table 4-4 shows the PROFIBUS-DP/V1 data types and their coding.

Table 4-2. PROFIBUS compared to Advant OCS Data Types

PROFIBUS DP/V1 Data Type	**PROFIBUS Number of Bytes**	**Advant OCS Data Type**	**Remarks**
Boolean	1	B	
Integer8	1	I, IL, B	
Integer16	2	I, IL, B	
Integer32	4	IL, B	
Unsigned8	1	I, IL, B	Transmitted as a byte in PROFIBUS.
Unsigned16	2	IL, B	Transmitted as two bytes in PROFIBUS.
Unsigned 32	4	IL, B	When the range is exceeded, the output is set to the maximum IntegerLong value.
Floating Point	4	R	Floating point according to IEEE STD 754
Visible String	1, 2, 3 ...	I, IL, B	Each byte is mapped separately.
Octet String	1, 2, 3 ...	I, IL, B	Each byte is mapped separately.

Table 4-3. Advant OCS compared to PROFIBUS Data Types

Advant OCS Data Type	**PROFIBUS DP/V1 Data Type**	**PROFIBUS Number of Bytes**	**Remarks**
B	Boolean	1	
I, IL, B	Integer8	1	Only the lower 8 bits of I and IL are used.
I, IL, B	Integer16	2	Only the lower 16 bits of IL are used.
IL, B	Integer32	4	
I, IL, B	Unsigned8	1	Only the lower 8 bits of I and IL are used.
I, IL, B	Unsigned16	2	Only the lower 16 bits of IL are used.
IL, B	Unsigned32	4	
R	Floating Point	4	
I, IL, B	Visible String	1, 2, 3 ...	Only the lower 8 bits of I and IL are used. Each byte is mapped separately.
I, IL, B	Octet String	1, 2, 3 ...	Only the lower 8 bits of I and IL are used. Each byte is mapped separately.

Table 4-4. PROFIBUS-DP/V1 Data Types and their Coding

PROFIBUS DP-V1 Data Type	Code	Number of Bytes
Boolean	1	1
Integer8	2	1
Integer16	3	2
Integer32	4	4
Unsigned8	5	1
Unsigned16	6	2
Unsigned32	7	4
Floating Point	8	4
Visible String	9	1, 2, 3 ...
Octet String	10	1, 2, 3 ...
Date (1)	-	-
Time of Day (1)	-	-
Time Difference (1)	-	-

(1) **Not** supported.

Examples on special identifier format PROFIBUS DP /V1 and data types:

Example 1 (Figure 4-12).

									Meaning:
Identifier	0	1	0	0	0	0	1	0	Input, 2 bytes Manufacturer-specific format
Length byte 1	1	0	0	0	0	1	0	0	Entire length consistency, 5 bytes input
Length byte 2	0	0	0	0	1	0	0	0	Data type Floating Point
Length byte 3	0	0	0	0	0	1	0	1	Data type Unsigned8

Figure 4-12. Example 1 of Special Identifier Format DP/V1 and Data Types

Example 1 shows the special identifier format DP/V1 for an analog input block containing a value (Floating Point) and status (Unsigned8).

Example 2 (Figure 4-13).

									Meaning:
Identifier	1	1	0	0	0	1	0	0	Output/Input, 4 bytes Manufacturer-specific format
Length byte 1	1	0	0	0	0	1	0	0	Entire length consistency, 5 bytes output
Length byte 2	1	0	0	0	0	1	0	0	Entire length consistency, 5 bytes input
Data type byte 3	0	0	0	0	1	0	0	0	Data type byte 3 Floating Point
Data type byte 4	0	0	0	0	0	1	0	1	Data type byte 4 Unsigned8
Data type byte 5	0	0	0	0	1	0	0	0	Data type byte 5 Floating Point
Data type byte 6	0	0	0	0	0	1	0	1	Data type byte 6 Unsigned8

Figure 4-13. Example 2 of Special Identifier Format DP/V1 and Data Types

Example 2 show the special identifier format PROFIBUS DP/V1 for an analog output block where the output value can be read back. The representation of this analog output block is structured as follows:

- Output value (Floating Point)
- Output status (Unsigned8)
- Read back value (Floating Point)
- Read back status (Unsigned8)

4.3 Receive Data

This section is a description on how to receive data from PROFIBUS slave. To be able to receive data you need a PB-R PC element.

Understanding PB-R PC Element:

PB-R PC element is used to access input data from a PROFIBUS slave. The address of the PROFIBUS slave is found in the corresponding data base element which instance reference is given by the **DBINST** terminal.

The PB-R PC element reads a consecutive sequence of data values (of the same type) from the data message received from the slave. The PB-R PC element puts read data on the **VALUE1**...**VALUE32** terminals and current data status on the **VALID** terminal. **VALID** is set TRUE if received data are valid. The number and type of data read, depends on the parameters C1 and C2.

Create a PB-R PC element, with parameters C1, C2, and C3, where C1 is the data type (B, I, IL and R) and C2 is the number of outputs (1...32). C3 (0...1) specifies if inputs (= 0) or outputs (= 1) shall be read. Reading outputs can be of interest when reading values from a slave being configured and controlled by another PROFIBUS master device (see READONLY terminal in the PBS data base element). You should get a view like Figure 4-14 on the screen:

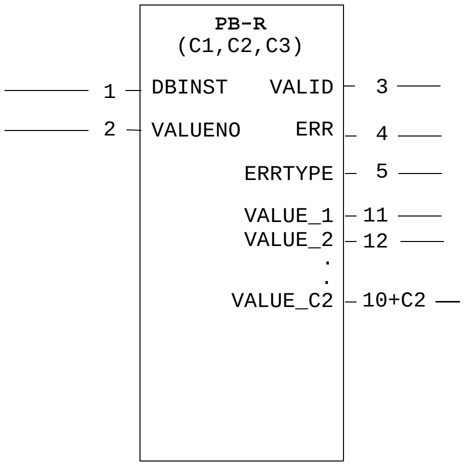


Figure 4-14. PB-R PC Element

In Figure 4-14 above you can see an overview over the PB-R PC element. In next section (Section 4.3.1, Configuration of PB-R PC Element) we will guide you through the configuration of this PC element.

4.3.1 Configuration of PB-R PC Element

This section is intended for you who want to start up the communication link with only the necessary information or feel a little bit unsure of how to configure a PB-R PC element. If you want more information, we recommend you to (also) read *PC Elements Advant Controller 400 Series Reference Manual* which also contains a complete list over the terminals discussed in this section.

In this section you will configure the PB-R PC element correctly to be able to access **input** data from a PROFIBUS slave.

We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **DBINST**. Input defining the corresponding PROFIBUS slave data base element (that is, a node on the network).
2. Look up your device's code number (often on the package or directly placed on the device), and look for it in the GSD file with GSD keyword: *Module*. Calculate how data is transferred from the device. This is described in Section 4.2, Understanding VALUENO and Identifier Formats.
3. Enter **VALUENO**. An identifier of the data to be read. See example below and Section 4.2.1, Understanding the VALUENO Terminal.

Example of a configured PB-R PC element.

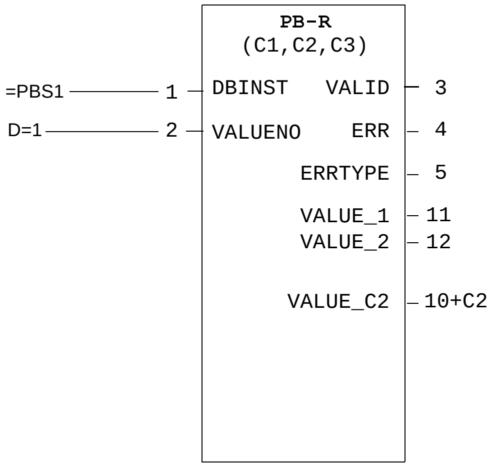


Figure 4-15. Example of a configured PB-R PC Element

In Figure 4-15 above a PB-R PC element has been configured in the following way:

- **DBINST** = PBS1, which means it is set to point to PROFIBUS slave data base element with terminal **NAME** set to PBS1.
- **VALUENO** = 1, which means we are interested in process value number 1 (see Section 4.2.1, Understanding the VALUENO Terminal for explanation of process values).

4. You have finished configuring the PB-R PC element. Move on to the PB-S PC element in Section 4.4, Send Data.

4.4 Send Data

This section is a description on how to send data from PROFIBUS slave. To be able to send data you need a PB-S PC element.

Understanding PB-S PC Element

PB-S PC element is used to output data to a PROFIBUS slave. The address is found in the corresponding data base element which instance reference is given by the **DBINST** terminal.

The element transfers data from the **VALUE1**...**VALUE32** terminals to a consecutive sequence of data values in the data message sent to the slave.

NOTE

More than one PB-S elements can update the same slave data value. There is no checking that the combination of the **VALUENO** terminal and C2 gives non-overlapping positions in the Payload Data Unit.

Create a PB-S PC element, with parameters C1 and C2, where C1 is the data type (B, I, IL or R) and C2 is the number of terminals (1...32). You should get a view like Figure 4-16 on the screen.

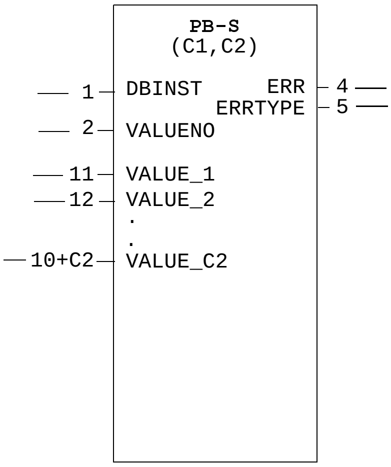


Figure 4-16. PB-S PC Element

In Figure 4-16 above you see an overview over the PB-S PC element. In next section (Section 4.4.1, Configuration of PB-S PC Element) we will guide you through the configuration of this PC element.

4.4.1 Configuration of PB-S PC Element

This section is intended for you who want to start up the communication link with only the necessary information or feel a little bit unsure of how to configure a PB-S PC element. If you want more information, we recommend you to (also) read *PC Elements Advant Controller 400 Series Reference Manual* which also contains a complete list over the terminals discussed in this section.

In this section you will configure the PB-S PC element correctly to be able to **output** data to a PROFIBUS slave.

We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **DBINST**. Input defining the corresponding PROFIBUS slave data base element (that is, a node on the network).
2. Look up your device's code number (often on the package or directly placed on the device), and look for it in the GSD file with GSD keyword: *Module*. Calculate how data is transferred from the device. This is described in Section 4.2, Understanding VALUENO and Identifier Formats.
3. Enter **VALUENO**. An identifier of the data to be written. See example below and Section 4.2.1, Understanding the VALUENO Terminal.

4. Enter **VALUEx**. How many values to enter depends on how you have configured parameter C2. These values could be of three different types:
   - Data. For example the number 6 or 78 or 5436...
   - Another PC element.
   - Data base element.

   Example of a configured PB-S PC element.

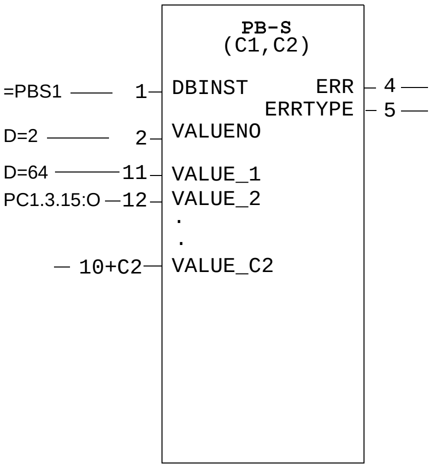


Figure 4-17. Example of a configured PB-S PC Element

In Figure 4-17 above a PB-S PC element has been configured in the following way:

- **DBINST** = PBS1, which means it is set to point to PROFIBUS slave data base element with terminal **NAME** set to PBS1.
- **VALUENO** = 2, which means we are interested in process value number 2 (see Section 4.2.1, Understanding the VALUENO Terminal for explanation of process values).
- **VALUE_1** = 64, which means we transfer data with value 64.
- **VALUE_2** = PC1.3.15:O, which is a connection to another PC element.

5. You have finished configuring the PB-S PC element. Move on to the PB-DIAG PC element in Section 4.5, Receive Diagnostic Data.

4.5 Receive Diagnostic Data

This section is a description on how to receive diagnostic data from PROFIBUS slave. To be able to receive diagnostic data you need a PB-DIAG PC element.

Understanding PB-DIAG PC Element

The PB-DIAG PC element is used to access diagnostic data from a PROFIBUS slave. The address of the PROFIBUS slave is found in the corresponding data base element which instance reference is given by the **DBINST** terminal.

The PB-DIAG PC element reads a diagnostic data value from the diagnostic data message received from the slave. The PB-DIAG PC element puts read data on the **VALUE1**...**VALUE32** terminals and current data status on the **VALID** terminal. **VALID** is set TRUE if data are valid. The number of outputs depend on the parameter C1 (1...32).

Create a PB-DIAG PC element, with parameter C1, where C1 is the number of outputs. You should get a view like Figure 4-18 on the screen:

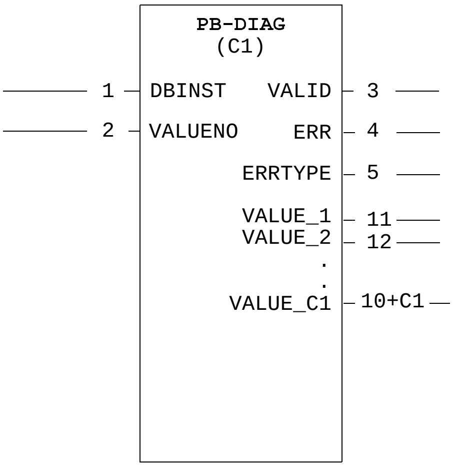


Figure 4-18. PB-DIAG PC Element

In Figure 4-18 above you see an overview over the PB-DIAG PC element. In next section (Section 4.5.1, Configuration of PB-DIAG PC Element) we will guide you through the configuration of this PC element.

4.5.1 Configuration of PB-DIAG PC Element

This section is intended for you who want to start up the communication link with only the necessary information or feel a little bit unsure of how to configure a PB-DIAG PC element. If you want more information, we recommend you to (also) read *PC Elements Advant Controller 400 Series Reference Manual* which also contains a complete list over the terminals discussed in this section.

In this section you will configure the PB-DIAG PC element correctly to be able to **receive** diagnostic data from a PROFIBUS slave.

We recommend that you follow the step-by-step guiding below, but it is not mandatory.

1. Enter **DBINST**. Input defining the corresponding PROFIBUS slave data base element (that is, a node on the network).
2. Look in device manual to figure out the type of diagnostics the device can send.
3. Enter **VALUENO**. A identifier of the data to be read. This number identifies the diagnostics data byte number. See example below.

Example of a configured PB-DIAG PC element.

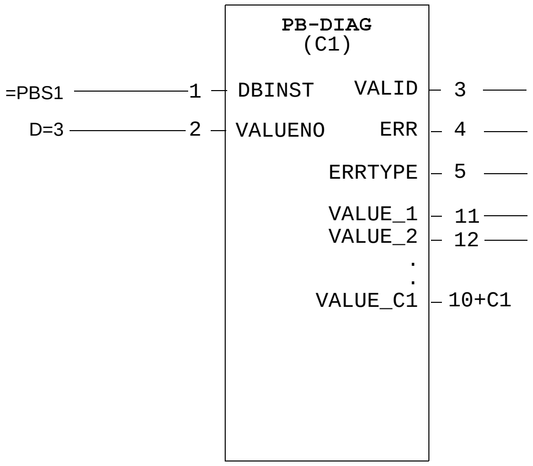


Figure 4-19. Example of a configured PB-DIAG PC Element

In Figure 4-19 above a PB-DIAG PC element has been configured in the following way:

- **DBINST** = PBS1, which means it is set to point to PROFIBUS slave data base element with terminal **NAME** set to PBS1.
- **VALUENO** = 3, which means we are interested in diagnostics data byte 3.

4. You have finished configuring the PB-DIAG PC element.

Chapter 5 Fault Tracing

This chapter describes the faults that might occur in an PROFIBUS network, how to diagnose and trace them, and what you can do to repair them.

For each one of the errors that may be reported by the communication interface, it is described what the error means, what might have caused the error, and how to repair the error when it is possible.

Remember that you often can discover the same error in four different ways:

- On LED's as hardware indicators
- On data base elements
- On PC elements
- And as system messages.

In the following sections we take a look at each one of them.

5.1 To do First

In this section we give you some advice on what to check first when an error has occurred.

If you have followed the instructions in Chapter 2, Installation and Chapter 3, Configuration and Chapter 4, Application Building, and still have a communication link that is not working properly, please check the following:

1. Is the cabling correctly set up? See Chapter 2, Installation.
   - Termination resistors must be properly in place.
   - Check whether the PROFIBUS connectors are correctly wired.
   - Are modems and cables installed as described in *Advant OCS Installation Rules User's Guide* and *Advant Controller 450 User's Guide*. Pay particularly attention to the power supply and connection to signal ground.
2. Is the PROFIBUS data base element correctly configured? See Chapter 3, Configuration.
   - All network numbers in the Advant controller 400 Series node must be unique.
3. Are the I/O device data base elements correctly configured? See Chapter 3, Configuration.
   - PROFIBUS characteristics must match the vendor's device description.
4. Check the data base and PC program application. See Chapter 3, Configuration and Chapter 4, Application Building
   - Is the CI541V1 submodule correctly configured? The green LED must be lit.
   - Are the PROFIBUS I/O devices correctly configured? The indicators of the PROFIBUS I/O devices must show "operating mode".

5. Are there any configuration errors reported?
   - Error messages may occur during download of the application, indicating that the configuration of the modules was not possible. Check the system messages and see Section 5.5, PROFIBUS related System Messages.
6. Have you made any significant changes in your configuration?
7. Have you made any other observations related to the problem?

A bus analyzer for monitoring the PROFIBUS communication link is also useful for further analysis. This test tool shows the traffic on the PROFIBUS communication link and can be used to detect problems.

5.2 Hardware Indicators

The communication interface CI541V1 has the following indicators on the front:

- The green LED is lit when the bus is operating.
- The red LED is lit when an error is detected on the CI541V1 submodule. The LED is also turned on at a reset of the submodule when the CI541V1 submodule is in selftest phase.

5.3 Data Base Elements Indicators

In this chapter we describe possible faults indicated in PROFIBUS data base elements. In Section 5.3.1, CI541 Data Base Element Indicators we discuss faults indicated in the CI541 data base element and in Section 5.3.2, PROFIBUS Slave Data Base Element Indicators we discuss faults indicated in the PROFIBUS slave data base element.

5.3.1 CI541 Data Base Element Indicators

First we take a look at the three terminals involved in fault indication on the CI541 data base element. Then we discuss what actions you can do to try to repair the error. The three terminals are described in Table 5-1.

Table 5-1. Error indicating Terminals on CI541 Data Base Element

Terminal No	Terminal Name	Default Value	PC Connection Data Type	Description
38	**WARNING**	-	B(r)	Warning flag indicating non-fatal errors.
39	**ERR**	-	B(r)	Error flag indicating fatal errors.
97	**DIAG**	-		Diagnostics information. See Table 5-2. Contains more information when the warning or error flags indicate an error.

In Table 5-2 you can see what has caused an error, and what you can do to repair the error. Error with highest priority first in the table.

Table 5-2. Diagnostics handling on CI541 Data Base Element

ERR	WARNING	DIAG	Condition	Action
1	-	**MNA**	Missing CI541V1. Device not accessible.	Check CI541V1 submodule.
1	-	**IE**	Internal software error.	Toggle the **IMPL** flag.
1	-	**ME**	Device error.	Toggle the **IMPL** flag. If error remains replace CI541V1.
1	-	**CE**	Configuration failed.	Change configuration parameters.
0	1	**SE**	System error.	The reason for this can be of a number of things. See system message for more detailed information.
0	1	**PE**	Process error.	The reason for this error condition can be disturbance on PROFIBUS or a physical error on CI541V1.
-	-	**PSV**	CI541V1 passive.	Set the **IMPL** flag.

5.3.2 PROFIBUS Slave Data Base Element Indicators

First we take a look at the four terminals involved in fault indication on the PROFIBUS slave data base element. Then we discuss what actions you can do to try to repair the error. The four terminals are described in Table 5-3. Slave behavior depends on every specific slave. For more information see the specific slave manual.

Table 5-3. Error Indicator Terminals on PROFIBUS Slave

Terminal No	Terminal Name	Default Value	PC Connection Data Type	Description
36	**WARNING**	-	B(r)	Warning flag indicating non-fatal errors.
37	**ERR**	-	B(r)	Error flag indicating fatal errors.
31	**DIAG**	-	I(r)	Diagnostics information. See Table 5-4. Contains more information when the warning or error flags indicate an error.
88	**SDIAGIND**	-	B(r)	PROFIBUS slave diagnostics data indication. Indicates that corresponding PROFIBUS slave has published diagnostics data. This data can be accessed with the PB-DIAG PC element.

In Table 5-4 you can see what has caused an error, and what you can do to repair the error. Error with highest priority first in the table.

Table 5-4. Diagnostics handling on PBS Data Base Element

ERR	WARNING	DIAG	Condition	Action
1	-	**FBE**	Fieldbus interface error.	Check CI541 data base element for further analysis.
1	-	**SNA**	Station not available.	Add slave to bus.
1	-	**CON**	Connection error.	Check cable and device. Toggle **IMPL** flag, on CI541 data base element.
1	-	**CE**	Configuration error.	Check configuration in data base element.
1	-	**IE**	Internal software error.	Toggle **IMPL** flag, on CI541 data base element.
1	-	**CA**	Connection aborted.	Reset slave device.
1	-	**DD**	Device deactivated.	Activate or reset slave device.
0	1	**SE**	System error.	The reason for this can be of a number of things. See system message for more detailed information.
0	1	**PE**	Process error.	See system message for more detailed information.
-	-	**PSV**	Slave passive.	Set the **IMPL** flag.
-	-	**OK**	Slave OK.	Slave OK and in operation.

5.4 PC Element Indicators

In Table 5-5 we describe possible faults indicated in PROFIBUS PC elements. They are found on the **ERRTYPE** terminal.

Table 5-5. Error Code handling on PC Element

Error Code	Condition
0	OK. No error.
1	Data base element according to **DBINST** not found.
2	Data base element not properly initiated.
3	Data base element error marked.
4	Bus error. The PROFIBUS interface module is not reachable.
5	The data base element is marked 'Not implemented'.
6	The data base element is marked 'Not in service'.

5.5 PROFIBUS related System Messages

As we mentioned earlier in this chapter, many of the errors can be detected in several ways. The source that gives you absolutely most information is system messages. In this section we describe and give examples on PROFIBUS related system messages.

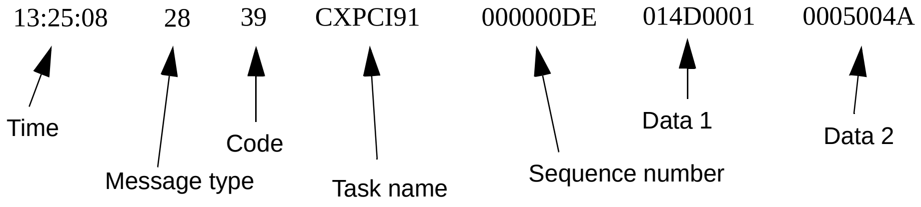


Figure 5-1. PROFIBUS related System Messages

Explanation on the parts of the system message in Figure 5-1 above:

- PROFIBUS related system messages always has *message type* 28.
- *Sequence number* - contain code specific information.
- *Data 1* - always consists of the logical file number 14D, for CI541V1, 14E for a PROFIBUS slave, followed by the logical record number.
- *Data 2* - contains MSW = bus number and LSW = station number, in some cases additional error information.

Below in Table 5-6 follows descriptions of the different types of error *codes*, what has caused the error and what you can do to try to repair the error.

Table 5-6. System Message handling on PROFIBUS related System Messages

Code	Condition	Action
20	CI541V1 failure.	See **NOTE** on page 5-9. 1. Remove and insert CI541V1 again. 2. Replace CI541V1.
21	Slave or CI541V1 failure. To find out which it is, see Data 1.	If error reported on slave: two possibilities: replace bus cable or replace slave. If error reported on CI541V1: see error code 20.
39	OK. Previously reported error is now cleared.	
-1	CI541V1 failure.	See **NOTE** on page 5-9. 1. Remove and insert CI541V1 again. 2. Replace CI541V1.
-4	System error.	If Data 2 is set *H'20000000*, then you should check if two or more devices has the same address. Any other values on Data 2 is additional diagnostics info.
-6	Communication error. Internal error on the communication module.	Restart system.

NOTE

Insertion of the CI541V1 **cannot** be done in a running Advant Controller 410. You first have to power off the Advant Controller 410, insert the CI541V1 and then turn on the power again.

In Advant Controller 450 you first have to unplug the submodule carrier (SC510 or SC520). After that you can insert the CI541V1 into this submodule carrier and finally plug in the submodule carrier again. This means that you do not have to power off the whole Advant Controller 450.

Appendix A Technical Data

A.1 Hardware Module

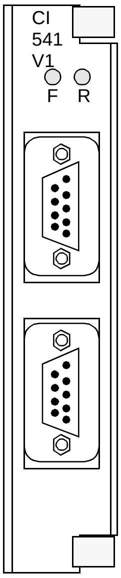


Figure A-1. CI541V1

Description

CI541V1 is a submodule destined to the submodule carrier SC510 and SC520 in Advant Controller 450 or PM150 in Advant Controller 410.

The communication channel supports communication speed up to 12 Mbps.

NOTE

Insertion of the CI541V1 **cannot** be done in a running Advant Controller 410. You first have to power off the Advant Controller 410, insert the CI541V1 and then turn on the power again.

In Advant Controller 450 you first have to unplug the submodule carrier (SC510 or SC520). After that you can insert the CI541V1 into this submodule carrier and finally plug in the submodule carrier again. This means that you do not have to power off the whole Advant Controller 450.

Technical Data

Indicators

LED R, Run (green) on module front. Indicating module running normally.

LED F, Fault (red) on module front. Indicating a fatal error detected on the CI541V1 module. the LED is also turned on at reset of the module.

Connectors

Serial channel 1 connector:

Connector type: 9-pin male D-SUB

Placement: on module front.

Pin assignment: See Table A-1 below.

Table A-1. Standard Pin Assignment of 9-pin SUB-D Connector

Pin number	Signal	Significations
1 (1)	-	Shield, Protected Ground
2 (1)	-	Minus 24V Output Voltage
3	RxD/TxD-P	Receive/Transmit-Data-P
4	CNTR-P	Control-P
5	DGND	Data Ground
6	VP	Voltage-Plus
7 (1)	-	Plus 24V Output Voltage
8	RxD/TxD-N	Receive/Transmit-Data-N
9	CNTR-N	Control-N

(1) Not used

Mechanical Data

Module size: The module occupies one submodule slot.

A.2 Performance

This section is a short description of CI541V1 submodule performance. The section contains calculation formulas and tables based on the mentioned formulas. Please observe that the tables do not contain any measured values, only calculated values.

System Reaction Time

The system reaction time depends mainly on the number of stations, the number of bytes per DP-Slave and the number and types of masters in the system. Furthermore the maximum reaction time (max Tsdr) of the individual stations will influence the system reaction time. In systems with several masters and slaves the system reaction time equals the Target Rotation Time.

Monomaster PROFIBUS-DP System

This configuration consists of one Master (class 1) and up to 125 DP-Slaves.

This section is a short description of the CI541V1 PROFIBUS-DP performance. The section contains calculation formulas and tables based on the mentioned formulas.

Approximate calculation[1] of required time for one bus cycle T_{BC}:

Formulas:

$T_{MC} = T_{S/R} + T_{SDR} \times T_{BIT} + T_{A/R} + T_{ID} \times T_{BIT} + T_{EXE} \times (N_I + N_O)$
$T_{S/R} = (N_{HD} + N_I) \times N_{BB} \times T_{BIT}$
$T_{A/R} = (N_{HD} + N_O) \times N_{BB} \times T_{BIT}$
$N_{IOA} = \Sigma\,(N_I + N_O)/N_{slave}$
$T_{BC} = T_{PRT} + \Sigma T_{MC}$

$$T_{BC} = 10\text{ ms} + [(198 + 2 \times MAXT_{SDR} + 11 \times N_{IOA})\,T_{BIT} + 0.0072\text{ms} \times N_{IOA}]\,N_{slave}$$

Definitions:

N_{HD} = 9 byte
N_{BB} = 11 bit/byte
T_{EXE} = 0.0072 ms
T_{PRT} = 10 ms
T_{SDR} = MAX T_{SDR}[2]
T_{ID} = MAX T_{SDR}

---

1. See also EN 50170
2. For worst case calculation

Abbreviations:

T_{BC} = Bus Cycle time PROFIBUS in ms
T_{MC} = Message Cycle time PROFIBUS per slave in ms
T_{PRT} = Program Run Time PROFIBUS interface in ms
T_{EXE} = Execution time data conversion PROFIBUS interface for one I/O value in ms
$T_{S/R}$ = Send/Request frame time in ms
T_{SDR} = Actual Station Delay of Responder in T_{BIT}
MAX T_{SDR} = Maximum Station Delay of Responder in T_{BIT}
$T_{A/R}$ = Acknowledgement/Response frame time in ms
T_{ID} = Idle time in T_{BIT}
T_{BIT} = Bit Time (1/baudrate)
N_{slave} = No of slaves
N_{IOA} = No of I/O bytes per slave
N_I = No of Input bytes per slave
N_O = No of Output byte per slave
N_{HD} = No of Header bytes per PDU
N_{BB} = No of Bits/Byte

Assumptions:

1. Mono master system (with one CI541V1).
2. All slaves are in data exchange mode, with no diagnostic data pending.
3. All slaves use simple identifier format. See Section 4.2.2, Understanding Identifier Formats.

The following figure shows the PROFIBUS configuration for calculation.

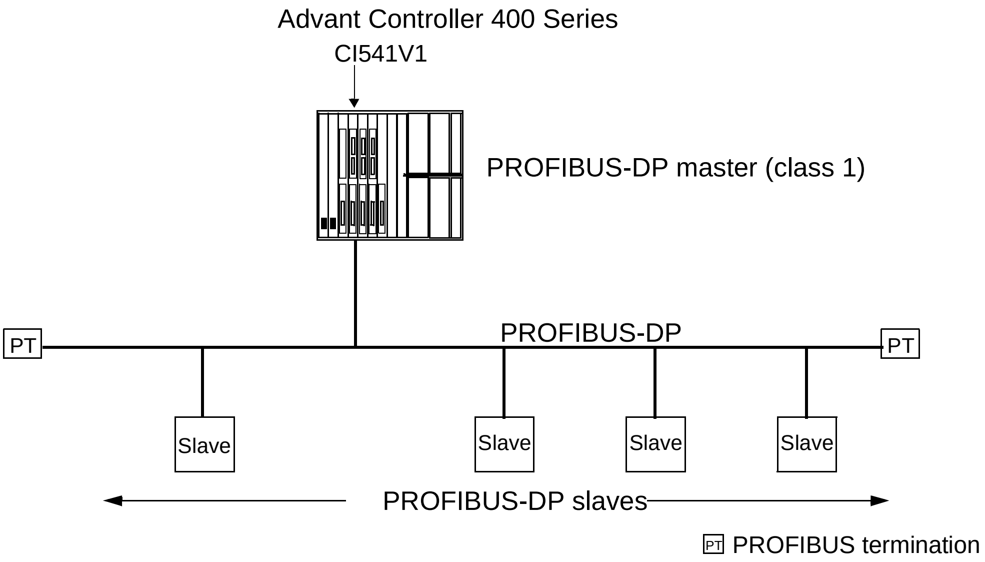


Figure A-2. PROFIBUS Configuration for Performance Calculation

The following tables show the different PROFIBUS cycle times depending on the transmission speed:

Table A-2. PROFIBUS Cycle Time based on Transmission Speed 93.75 kbit/sec

Slaves	**PROFIBUS cycle time [ms]**			
	Input/output bytes per slave			
	2/2 byte	16/16 byte	32/32 byte	244/244 byte
1	14	17	21	74
5	29	47	67	331
10	49	84	124	652
30	127	231	351	1935

.

Table A-3. PROFIBUS Cycle Time based on Transmission Speed 187.5 kbit/sec

Slaves	**PROFIBUS cycle time [ms]**			
	Input/output bytes per slave			
	2/2 byte	16/16 byte	32/32 byte	244/244 byte
1	12	14	16	44
5	20	29	40	179
10	30	48	69	348
30	69	124	187	1025

Table A-4. PROFIBUS Cycle Time based on Transmission Speed 500 kbit/sec

Slaves	**PROFIBUS cycle time [ms]**			
	Input/output bytes per slave			
	2/2 byte	16/16 byte	32/32 byte	244/244 byte
1	11	12	13	25
5	15	19	23	85
10	19	27	37	160
30	37	62	90	461

Table A-5. PROFIBUS Cycle Time based on Transmission Speed 1500 kbit/sec

Slaves	**PROFIBUS cycle time [ms]**			
	input/output bytes per slave			
	2/2 byte	16/16 byte	32/32 byte	244/244 byte
1	10	11	11	17
5	12	14	16	47
10	14	18	23	84
30	22	34	48	233

Multimaster PROFIBUS-DP System

This configuration consists of up to 125 devices on the PROFIBUS network. It can include PROFIBUS- Masters (class 1, class 2, FMS) and PROFIBUS-Slaves (DP, DPV1, FMS).

For multimaster systems the buscycle time calculation is very complex. In this case, the individual Token Hold Time (T_{TH}) of each DP-Master is part of the calculation. For a DP-Master the Token Hold Time is represented by a complete poll cycle with it's associated DP-Slaves. The buscycle time for a token round containing all traffic on the PROFIBUS network is represented in the Token Rotation Time (T_{TR}).

Target Rotation Time

According to EN50170, the following calculation for the Token Rotation Time takes place:

for FMS-Masters and DP-Masters (class 2):
$T_{TR(FMS)} > \Sigma T_{SLP}$

for DP-Masters (class 1):
$T_{TR(DP)} = T_{SLP(TS)} + T_{TR(FMS)} + T_{ADD}$

$T_{ADD} = (T_{ID1} + T_{SL} + 2*255*11*t_{BIT})*(Max\_Retry\_Limit + 1)$

Legend:
$T_{TR(FMS)}$ = FMS- and DP-Masters (class 2) Target Rotation Time
ΣT_{SLP} = Sum of the poll cycles of all DP-Masters (class 1)
$T_{SLP(TS)}$ = Poll cycle of the DP-Master (class 1) "TS"
$T_{TR(DP)}$ = The DP-Masters (class 1) Target Rotation Time
T_{ADD} = Safety margin for message retransmissions

For CI541 the approximate calculation of required time for one bus cycle T_{BC} can be used for $T_{SLP(TS)}$. The time T_{PRT} can be neglected if the real Target Rotation Time is greater than $T_{PRT.}$ Refer to the documentations of the masters for the calculations of their individual $T_{SLP(TS)}$.

CI541 acting as Master Class 2

CI541 is capable to act as a class 2 Master with the PROFIBUS commands Read_Input, Read_Output and Slave_Diagnostic. Slave_Diagnostic is handled as a slow background cycle and not observed for the calculations. To read inputs and/or outputs, CI541 is able to insert one acyclic data transfer per poll cycle. So a not owned slave with inputs and outputs takes two T_{TR} to finish. For updating all configured not owned slaves, the following approximate calculation can be used:

$T_{SRnoslave} = 2 \times T_{TR} \times N_{noslave} + T_{acyc}$

For the T_{BC} calculation of CI541the time of the acyclic data transfer must be considered. If the T_{BC} for the CI541 itself is less than 65 ms it should be assumed as 65 ms for the calculations of $T_{SRnoslave}$. For simplification, the time for acyclic data tranfer (T_{acyc}) is calculated separately and for T_{TR} calculation the T_{BC} as described under term Multimaster PROFIBUS-DP System is used.

$T_{acyc} = (220 + 2 \times MAXT_{SDR} + 11 \times N_{IOAacyc}) \times T_{BIT} \times N_{noslave}$

A acyclic message cycle is defined as:

$T_{MCacyc} = T_{S/Racyc} + T_{SDR} \times T_{BIT} + T_{A/R} + T_{ID} \times T_{BIT}$
$T_{S/Racyc} = N_{HDacyc} \times N_{BB} \times T_{BIT}$
$T_{A/R} = (N_{HD} + N_{IO}) \times N_{BB} \times T_{BIT}$
$N_{IOAacyc} = \Sigma\ (N_{Iacyc} + N_{Oacyc})/N_{noslave}$
$N_{noslave} = N_{Inoslave} + N_{Onoslave}$
N_{HDacyc} = 11 byte

Abbreviations:

$T_{S/Racyc}$ = Acyclic Send/Request frame time in ms
$T_{SRnoslave}$ = System reaction time for update cycle of a not owned slave
N_{HDacyc} = No of Header bytes for acyclic requests
N_{IO} = Number of input or output bytes to read from not own slave
$N_{noslave}$ = No of not owned slave poll requests
$N_{Inoslave}$ = No of not owned slaves handled with read inputs
$N_{Onoslave}$ = No of not owned slaves handled with read outputs

Assumptions:

4. Diagnostic data transfer neclected.
5. All slaves are in data exchange.

Appendix B PROFIBUS-DP

B.1 PROFIBUS-DP Technical Overview

This appendix is a description over PROFIBUS-DP functions and characteristics. Most of the facts are taken from **www.profibus.com**. PROFIBUS-DP is designed for high-speed data communication at the device level. Here, central controllers (that is, PLCs/PCs) communicate with their distributed field devices (I/O, drives, valves, and so on) via a high-speed serial link.

Basic PROFIBUS-DP Functions

The central controller (Master) cyclically reads the input information from the slaves and cyclically writes the output information to the slaves. The bus cycle time must be shorter than the program cycle time of the central PLC which for many applications is approximately 10 msec. In addition to cyclic user data transmission, PROFIBUS-DP provides powerful functions for diagnostics and configuration. Data communication is monitored by monitoring functions on both the master and slave side.

Transmission Technology (See more in Data Transmission)

- RS-485, twisted pair, two wire cable or fibre optics.
- Baud rates from 9.6 kbit/sec to 12 Mbit/sec.

Bus Access

- Token passing procedure between masters and master-slave procedure for slaves.
- Mono-master or multi-master systems possible.
- Master and slave devices, maximum of 126 stations on one bus.

Communication

- Peer-to-peer (user data transmission) or Multicast (control commands).
- Cyclic master-slave user data transmission and acyclic master-master data transmission.

Operating Modes

- Operate: Cyclic transmission of input and output data.
- Clear: Inputs are read, and outputs are held in fail-safe status.
- Stop: Only master-master data transmission is possible.

Synchronization

- Control commands permit synchronization of the inputs and outputs.
- Sync mode: Outputs are synchronized.
- Freeze mode: Inputs are synchronized.

Functionality

- Cyclic user data transmission between DP master and DP slave(s).
- Dynamic activation or deactivation of individual DP slaves.
- Check of DP slave configuration.
- Powerful diagnostic functions, 3 hierarchical levels of diagnostic messages.
- Synchronization of the inputs and/or the outputs.
- Address assignment for the DP slaves over the bus.
- Configuration of the DP master (class 1) over the bus.
- Maximum of 244 bytes of input and output data per DP slave.

Security and Protection Functions

- All messages are transmitted with Hamming distance HD = 4.
- Watchdog timer at the DP slave.
- Access protection for the inputs/outputs of the DP slaves.
- Monitoring of user data transmission with configurable monitoring timer at the master.

Types of Devices

- Class-2 DP master (class 2): programming/configuration/diagnostic device.
- Class-1 DP master (class 1): central programmable controllers such as PLCs, PCs, and so on.
- DP slave: device with binary or analog inputs/outputs, drives, valves, and so on.
- Peer-to-peer (user data transmission) or Multicast (control commands).
- Cyclic master-slave user data transmission and acyclic master-master data transmission.

Data Transmission

In PROFIBUS-DP the master addresses cyclically all connected slaves. It sends up to 244 bytes output data and receives up to 244 bytes of input data. Additionally there can be up to 244 bytes of diagnostic data from the slave. The transmission of diagnostic data is requested by the slave when it sets the *Slave_Diag* flag. The contents of the first six bytes in the diagnostic data Payload Data Unit are defined by the norm, EN 50170. The remaining 238 bytes may be assigned by the manufacturer of the PROFIBUS slave as it seems appropriate.

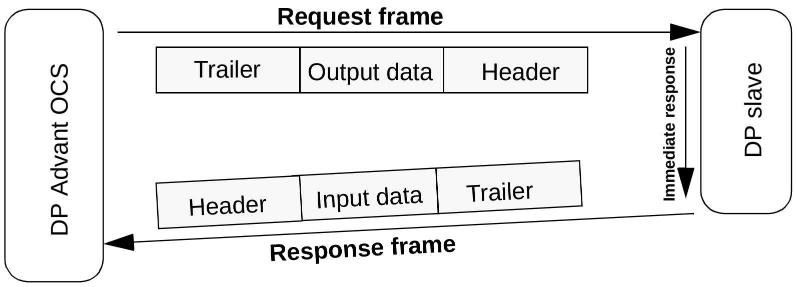


Figure B-1. PROFIBUS-DP Data Transmission

System Configuration and Types of Devices

PROFIBUS-DP permits mono-master or multi-master systems. Up to 126 devices (master or slaves) can be connected to one bus. Each PROFIBUS-DP system can contain three different types of devices:

DP Master (Class 1)

The class-1 DP master is a central controller which exchanges information with the decentralized stations (i.e., DP slaves) within a specified message cycle.

DP Master (Class 2)

Class-2 DP masters are programmers, configuration devices or operator panels. They are used during commissioning for configuration of the DP system or for operation and monitoring purposes.

DP Slave

A DP slave is a peripheral device (I/O devices, drives, HMI, valves, etc.) which collects input information and sends output information to the controller. There are also devices which supply only input or only output information.

The amount of input and output information depends on the device type. A maximum of 244 bytes of input information and 244 bytes of output information is permitted.

In mono-master systems only one master is active on the bus during the operating phase of the bus system.

In multi-master configurations several masters are connected to a bus. These masters are either independent subsystems each consisting of one master (class 1) and its assigned DP slaves, or additional configuration and diagnostic devices. The input and output images of the DP slaves can be read by all DP masters. However, only one DP master may write-access the outputs. Multi-master systems produce a longer bus cycle time than mono-master systems.

System Behavior

DP masters (class 1) can be controlled either locally or via the bus by the configuration device. There are three main states:

- Stop - In this state, no data transmission between the master and the DP slaves occurs.
- Clear - In this state, the master reads the input information of the DP slaves and holds the outputs in fail-safe status.
- Operate - In this state, the master is in the data transfer phase. In a cyclic data communication, inputs of the DP slaves are read, and output information is written to the DP slaves.

The master cyclically sends its local status to all DP slaves assigned to it with a Multicast command at a configurable time interval.

Protection Mechanisms

Time monitoring is provided for the DP master and the DP slaves. The monitoring interval is specified during configuration.

At the DP Master

The DP master monitors data transmission of the slaves with the Data_Control_Timer. A separate control timer is used for each slave. This timer expires when correct data transmission does not occur within the monitoring interval. The user is informed when this happens. If the automatic error reaction (Auto_Clear = True) has been enabled, the master exits its OPERATE state, switches the outputs of all assigned slaves to fail-safe status and changes to its CLEAR status.

At the DP Slave

The slave uses the watchdog control to detect failures of the master or the transmission line. If no data communication with the master occurs within the watchdog control interval, the slave automatically switches its outputs to the fail-safe status.

INDEX

B
Bus analyzer 5-2
Bus connector 2-5
Bus cycle A-3
Bus error 5-7

C
Cable length 2-2
Call names 3-2
CI541V1 failure 5-8
Communication error 5-8
Communication link 3-1
Configuration error 5-6
Connection error 5-6

D
Data transfer rate 2-2
Data type 4-11
Device error 5-4
Diagnostic data 4-1
Diagnostics
    CA 5-6
    CE 5-4, 5-6
    CON 5-6
    DD 5-6
    FBE 5-6
    IE 5-4, 5-6
    ME 5-4
    MNA 5-4
    OK 5-6
    PE 5-4, 5-6
    PSV 5-4, 5-6
    SE 5-4, 5-6
    SNA 5-6
Dimensioning data 3-3

E
EN 50 170 2-5
EN 50170 2-2 to 2-3
Error code 5-8

F
Fieldbus interface error 5-6

G
GSD file 3-2, 3-7, 3-11 to 3-13, 4-16, 4-18
GSD files 4-4

H
Hardware indicators 5-2

I
Input data 4-1, 4-15
Internal software error 5-4, 5-6

L
LED 2-10, 5-1 to 5-2

M
Media 2-3
Message sending 4-2
Modular devices 3-13
Module byte 4-5

N
Node numbers 3-1

O
Output data 4-1, 4-11, 4-18

P
Payload Data Unit 4-2, 4-17
Performance A-4
Process error 5-4, 5-6
PROFIBUS Library Editor 3-1, 4-1

R
Repeaters 2-2

S
Segment 2-2
Sequence number 5-8
System error 5-4, 5-6, 5-8
System message 5-1 to 5-2, 5-7

T

Terminals 3-17, 5-3, 5-5
  BUS 3-2, 3-17
  BUSNO 3-5
  CFGD1 3-12 to 3-13
  CFGD25 3-12 to 3-13
  CFGDLEN 3-12 to 3-13
  DBINST 4-15 to 4-19, 4-21, 5-7
  DIAG 5-3, 5-5
  DP_V1 3-8
  ERR 5-3, 5-5
  FAILSAFE 3-8
  IDENTNO 3-8
  IMPL 3-2, 3-5, 3-16, 5-4, 5-6
  MAXDIAGL 3-8
  MINSINT 3-9
  POSITION 3-5
  REFDESCR 3-17
  SDIAGIND 5-5
  STATION 3-2, 3-17
  STNNO 3-5
  SUBPOS 3-5
  UPRMD1 3-10 to 3-11
  UPRMD25 3-10 to 3-11
  UPRMDLEN 3-10 to 3-11
  VALID 4-15, 4-20
  VALUE1 4-15, 4-17
  VALUE32 4-15, 4-17
  VALUENO 4-1 to 4-2, 4-4, 4-16 to 4-19, 4-21
  WARNING 5-3, 5-5
  WDFACT1 3-8
  WDFACT2 3-8
Token 1-4
Transmission speed 3-1
Transmitters 1-4

3BSE 016 040R101 Rev A

March 2000